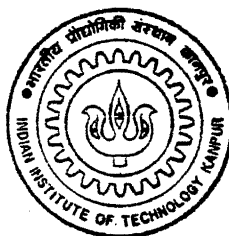# A Finite Volume Solution Method for Fluid Flow Problems

by

Atul Kumar Verma

DEPARTMENT OF MECHANICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

MAY, 1997

A Finite Volume Solution Method for Fluid Flow Problems

A Thesis Submitted

in Partial Fulfilment of the Requirements

for the Degree of

Doctor of Philosophy

by

Atul Kumar Verma

to the

DEPARTMENT OF MECHANICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

May, 1997

ME - 1997- D - VER - FIN

CERTIFICATE

It is certified that the work contained in the thesis entitled **A Finite Volume Solution Method for Fluid Flow Problems** by **Atul Kumar Verma**, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Dr. Vinayak Eswaran
Associate Professor
Department of Mechanical Engineering
Indian Institute of Technology Kanpur

May, 1997

Dedicated to -

My Parents

Acknowledgements

I take this opportunity to express my indebtedness and deep sense of gratitude to my thesis supervisor Dr. V. Eswaran for introducing me into the exciting field of CFD. His inspiring guidance, systematic approach, stimulating discussions and extensive care helped to shape my thesis as well as my personal attitude towards academics. It is my immense good fortune to be under his tutelage and I am profoundly grateful to him.

With gratitude and respect I thank Prof. Gautam Biswas for his keen interest in my academic progress and for extending excellent laboratory facilities for my work. I also thank Dr. B.S. Murty for the help I received from him especially in selecting test problems to validate the algorithm developed during the course of this programme.

I am grateful to my colleagues in the CFD laboratory, namely, Arun Saha, Dr. P.M.V. Subbarao, P.K. Maji, P. Bera, Satya Prakash, G. Jilani, Veera and Debashish for help I received from them on various occasions.

I extend my profound thanks to my friends Shrinivas, Ajay, Dr. Subbarao, Ravindra, Manish, Pramod, Dr. Sanjay Verma, Dr. Jawed, Dr. Shamsi, Alok Sharan, Dr. S.C. Mishra, G.K. Singh, Amarendra, Dr. S.C. Pradhan, Dr. D. Murthy, Dr. K.S. Venkatesh, Dr. Sandeep Pandey and Dr. M. Jayateertha, who have made my stay here a most memorable one.

It is difficult to find words to express my gratitude to my fiancee Seema, my brothers Mridul, Mukul and sister Sonal for their infallible support and inspiration throughout my stay here.

This work could not have been completed without the blessings of my parents. Even to think of thanking them is to trivialise all that they have done for me. My profoundest debts to them therefore, remain silent and unacknowledged.

<div align="right">

Atul Kumar Verma
IIT Kanpur
May 1997

</div>

Synopsis

Name of Student: **Atul Kumar Verma**   Roll No. **9220561**

Degree for which submitted: **Ph.D.**   Department: **Mechanical Engineering**

Thesis Title:

A Finite Volume Solution Method for Fluid Flow Problems

Name of thesis supervisor: **Dr. Vinayak Eswaran**

Month and year of synopsis submission: **May 1997**

SYNOPSIS

In this thesis we present a calculation procedure for incompressible 2-D laminar flows in complex geometries, that was developed with the aim that it be accurate, efficient and robust. We develop a finite volume scheme applicable directly on structured non-orthogonal grids. The solver was developed with the following points in mind :

(i) The governing equations be solved directly on the physical domain rather than on a transformed domain. There are many difficulties associated with transformation which justify this approach. Among other advantages, the governing equations are simpler in the physical domain; furthermore, Cartesian components of velocity are the natural variables in this approach, and do not introduce additional complexity (associated with the covariant or contravariant) into the equations.

(ii) The method should be finite-volume, as this method has shown itself to often deliver a good compromise between geometric flexibility, accuracy, and algorithmic simplicity. The grid should be non-staggered (collocated) which is more natural for non-orthogonal grids.

(iii) Convective and diffusive terms be computed with second-order accuracy, at least on uniform grids. Most new methods avoid first-order discretizations which introduce dissipation error. We also have the complication that the discretization should be applicable to non-orthogonal grids.

(iv) A proper means of higher-order upwinding for the convection terms be implemented to ensure accuracy and stability. If second-order accuracy is to be preserved, as well as stability ensured, on a non-orthogonal grid a new approach has to be developed.

(v) A flux limiter be developed to suppress the spurious oscillations that are known to develop in second-order upwind schemes. Second-order schemes for convection often display *dispersion* errors, i.e., unphysical over-/undershoots in solutions with strong gradients. These errors are sought to be contained by *flux limiting*.

(vi) The solver be tested on steady-state as well as on transient convection-diffusion problems; and for transient problems, both explicit and implicit time-stepping be implemented. Given the many aspects of accuracy, stability, boundedness, etc., involved in convective term discretizations, new schemes for convection modelling are tested not on the cumbersome Navier-Stokes equations, but on the convection-diffusion equation which also has all these problems of convective and diffusive term discretization. This simpler equation allows us to focus directly on the problems of discretization and delink them from the other problems associated with the Navier-Stokes equations.

(vii) The solver should retain its accuracy over a large range of grid-Peclet numbers. For explicit schemes it should be numerically stable for Courant numbers upto unity, and in implicit schemes for Courant numbers significantly greater than unity.

(viii) The method should be applied to the Navier-Stokes equations. A major problem in Navier-Stokes equation discretization is that of *pressure-velocity decoupling*. If the same grid points are used for pressure and velocity, and no preventive measures are taken, the pressure-velocity iterations may not converge because the pressure and velocity fields decouple, i.e., changes in one do not necessarily communicate to the other. This can easily and naturally be prevented by using a staggered grid, in which pressure and velocity grid points have different locations. However if collocated grids are used, methods have to be found to prevent the above mentioned decoupling.

(ix) Each of the above features should be credibly demonstrated by through validation on test problems.

The goals (i)-(ix) set out above have substantially been met. The basic finite-volume scheme developed in this thesis is called *OCV* (after Over lapping Control Volume) and its properties are extensively tested on the convection-diffusion equation and it is then applied

to the Navier-Stokes equations. The OCV method uses an iso-parametric formulation to compute diffusion and to introduce higher order upwinding. However, it avoids the need for assembly, common in finite-element algorithms.

Chapter 2. The OCV Method for Convection-Diffusion problems

Most of the extant finite-volume methods avoid overlapping control volumes by defining and using intermediate points. The values of the variables on these intermediate points are often obtained by line-interpolation from the neighbours. On non-orthogonal grids, it may be preferable to use higher-order finite-element type shape-functions for interpolation. To do this it is convenient to avoid intermediate points and, by accepting overlapping control volumes, to use the neighbouring points directly in the computations. The finite-element method with iso-parametric formulation usually requires assembly of elemental matrices. In this work, we discretize the equations using finite-element type shape functions, while avoiding the need for assembly.

The solution domain is discretized into a structured non-orthogonal grid as shown in Figure 1. This can be done by any grid-generation package for boundary-fitted systems. A typical control volume is shown by the shaded area in the figure. This choice of control volume uses the grid point coordinates directly, without computing any intermediate points, to form control volumes. It can be seen that each interior grid-point has a control volume associated with it, of which it is the central node. Hence we can refer to these control volumes by the index of this central node, e.g., the control volume for (i, j) is shown in Figure 1. It can be seen that adjacent control volumes will *overlap* to some extent.

The integral form of the conservation form of the two–dimensional steady–state convection–diffusion equation for a scalar ϕ is

$$\oint_{cs} \phi(\rho u n_x + \rho v n_y) dl \;=\; \oint_{cs} \Gamma(\frac{\partial \phi}{\partial x} n_x + \frac{\partial \phi}{\partial y} n_y) dl + \int \int S_\phi dA$$

where ρ is the density, u and v are the components of the velocity vector in the directions x and y, respectively, Γ is the diffusion coefficient, S_ϕ is a source term, dl is an elemental length on the boundary (cs) of the control volume, n_x and n_y are the direction cosines of the outward normal $\hat{n}$ of dl. The contour integration is counter–clockwise.

Using the mid–point rule we approximate the convective term :

$$\oint_{cs} \phi(\rho u n_x + \rho v n_y) dl \;=\; \sum_{k=1}^{4} \phi^{(k)} (\rho u^{(k)} \Delta y^{(k)} - \rho v^{(k)} \Delta x^{(k)})$$
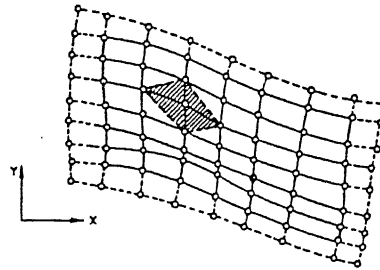
$$=\; \sum_{k=1}^{4} \phi^{(k)} F^{(k)}$$

Figure 1(a)

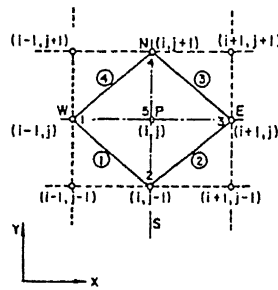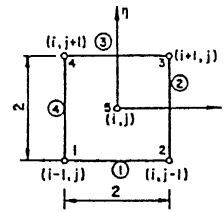Figure 1(b)

Figure 1(c)

Fig. 1. Schematic diagram of the computational domain

where the superscript (k) refers to the edges of the control volume (shown circled in Figure 1).

To incorporate upwinding, the scalar value $\phi^{(k)}$ is approximated at the mid-point of control-surface by interpolation within the appropriate control volume depending on the flow direction across that surface. This scheme for convection modelling is conservative and second-order accurate. The method used for interpolation is based on finite–element type shape functions. The iso-parametric formulation is used. The diffusion term is also approximated using the mid-point rule. The derivatives of the shape functions are evaluated at the mid-points, in (ξ, η) space, of the control surfaces. The summation is carried out over all the edges of the control volume.

A number of steady-state convection-diffusion problems have been solved by the OCV method and the results are compared with those of other schemes. In a typical case, we demonstrate that OCV, like QUICK, has second-order accuracy. We compare RMS
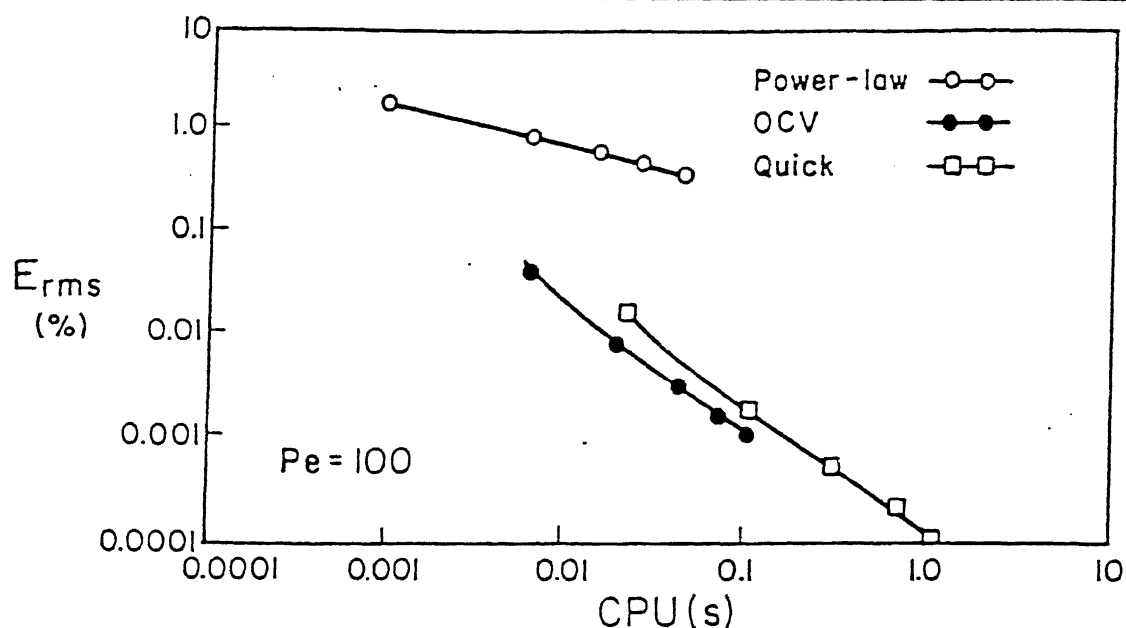
Fig. 2. RMS error versus CPU time

error Vs CPU time in Figure 2 for a purely convective test problem with a smooth solution. We find OCV, because it converges faster than QUICK, gives better accuracy for a given CPU time. The power-law performs relatively poor in comparison to the other two schemes. This observation was repeated in other test problems. Some other observations based on the results obtained by solving the convection-diffusion equation are summarized below.

- The results show that the OCV scheme performs well in cases when either convection or diffusion, or both, are significant. The results also show that the scheme reduces false-diffusion to a considerable extent in comparison with the power-law and other first-order schemes.

- The OCV treatment of diffusion seems to be very effective, even on distorted meshes. On uniform meshes, it is second-order accurate.

- The OCV treatment of the convection term, like that of the QUICK scheme, is second order accurate on an uniform mesh. Although the OCV scheme does not out-perform QUICK on the same grid size, it does better for the same CPU-time.

- On non-orthogonal grids, OCV gives better accuracy for a large and practical range of Peclet numbers than does QUICK applied to the transformed equations using the conventional five-point diffusion modeling, while it is computationly less expensive.

Chapter 3. A Bounded Convection Scheme for the OCV Method

We develop a flux limiting scheme *FLOCV* for the OCV method for 2-D steady and unsteady convection – diffusion problems on structured non-orthogonal grids. FLOCV switches from second to first-order interpolation in the presence of extrema. Smooth switching between the two is ensured by weighted averaging of second-order and first-order upwind differencing, with the weights being dynamically determined. The effectiveness of FLOCV is shown in a number of test problems.

We test the performance of FLOCV under varied and arduous conditions. The cases considered include steep gradients, flow-to-grid skewness and circulating flows on uniform and non-orthogonal structured grids. FLOCV performs well in all the test cases considered.

Table 1. Test problem 4 – Comparison of various flux limiters

Scheme	40 × 40			80 × 80		
	Max.	Min.	RMS	Max.	Min.	RMS
FOU	6.257	0.0	1.445	8.526	0.0	1.289
SOU	16.356	-3.58	1.268	18.880	-5.575	1.329
QUICK	18.808	-5.88	1.737	35.471	-21.092	3.366
MPL	9.973	0.0	0.936	10.0	0.0	0.717
MSOU	10.0	0.0	0.855	10.0	0.0	0.537
SHARP	10.219	-0.44	0.948	10.892	-1.373	0.652
FLOCV	9.992	0.0	0.755	10.0	0.0	0.5312

In a typical case, we present result for a stringent test problem in which a square-shaped scalar source field is advected diagonally across a square domain. The source field has a scalar value of 10 and the rest of the domain has a scalar value of 0. Tamamidis and Assanis have compared the performance of various schemes for this problem. These results are shown in Table 1. The *global* RMS error and minimum and maximum of the computed scalar field for the FLOCV is also shown in Table 1. It is clear that the performance of FLOCV compares well with that of other flux limiting schemes. The salient results are :

- For problems with discontinuities, FLOCV is effective in removing oscillations associated with the unbounded OCV scheme on both orthogonal and non-orthogonal grids.

- The effect of flux limiting on the accuracy of the base scheme was studied using a smooth Gaussian profile. It was demonstrated that FLOCV is second-order accurate on uniform Cartesian grids. On mildly non-orthogonal grids (up to 10 percent distortion) FLOCV remains second order accurate. The deterioration in the order-of-accuracy of the FLOCV is small on a moderately (20 percent) distorted grid.

- FLOCV is also applied to unsteady test cases. It smoothens waviness in the solution and decreases numerical spreading.

- The performance of FLOCV was judged with other standard flux limiting schemes and was comparable to the best of the other schemes.

Chapter 4. Application of the OCV Method to Solute Transport Problem

Modelling the transport of dissolved solutes in ground water flows of practical interest requires the numerical solution of a transient convection-dispersion equation in two, or more, dimensions. The numerical schemes for these equations need to have sufficient accuracy and also be adaptable to complex geometries, which are common in practical applications of solute transport problems. Most schemes in use on such problems have first-order accuracy. Integrations of first-order accuracy have diffusive errors which tend to spread the solution at sharp fronts. Therefore, second-order integration schemes are preferable for problems with such fronts and we present a second-order OCV scheme adapted to the solute transport problem. However, second-order schemes tend to produce solutions with spurious oscillations at the fronts and we explore methods such as flux-limiting and introduction of artificial diffusion to prevent these oscillations.

An overlapping control volume (OCV) technique is used for solving the two-dimensional, transient, solute-transport equation for ground water flows. Two-dimensional domains with orthogonal and non-orthogonal grids are considered. The solution with Crank-Nicolson and fully implicit time-stepping schemes are compared. We demonstrate the scheme's applicability for a wide range of Courant and grid Peclet numbers. In all cases, comparisons are done with known analytical solutions. We investigate the accuracy of different time-stepping schemes for the OCV method and discuss the circumstances under which spurious oscillations arise, and methods for their removal. The salient features are summarised below

- Implicit and Crank-Nicolson schemes are used for time stepping.

- The scheme is shown to work well for a wide range of grid Peclet numbers and Courant numbers well in excess of 1.

- Flux limiting is shown to be effective for Courant number less than unity, and the controlled introduction of artificial diffusion removes oscillations for higher Courant numbers.

- The effect on accuracy of moderate non-orthogonality of the grids is shown to be insignificant.

## Chapter 5. A Solution Method of Navier-Stokes Equation on Nonstaggered Grids

An OCV algorithm for computing steady and unsteady solutions to the two-dimensional incompressible Navier-Stokes equations on nonstaggered grids is presented. Since a nonstaggered arrangement is used in this formulation, the pressure-velocity decoupling or checkerboard pressure distribution may occur if the variables (velocities and pressure) at the cell edges are calculated by linear interpolation. We, following the *momentum interpolation* idea of Rhie and Chow, use a formulation in which the velocity at the cell faces are computed by allowing linear interpolation of the convective and diffusive terms but not of the pressure term. Although only steady-state cases are computed, the algorithm developed is for the unsteady problem and is time accurate. In this algorithm, we solve only for the pseudopressure not for the true pressure. (The pseudo-pressure is obtained by a rather simple homogeneous Neumann boundary condition instead of the complicated non-homogeneous and *implicit* boundary condition needed for the true pressure).

The calculation procedure is applied to variety of test problems. The steady-state solution is obtained by the false-transient approach. The two-dimensional laminar flow over a backward-facing step in a channel provides an excellent test case for the accuracy of numerical scheme because of the dependence of the reattachment length (Lr) on the Reynolds number. Excessive numerical diffusion, or smoothing in order to get stability, will result in failure to predict correct reattachment length. The results for the OCV method and that reported by Thompson and Ferziger (1989) have been presented in Table 2.

Other test cases considered are the flow in a driven square cavity, flow between concentric rotating cylinders, flow through a channel with sudden expansion, and vortex-shedding behind a square cylinder. The results of the square cavity problem are shown in Figure 3, with the results of Ghia and Ghia (1985). This algorithm works well on both orthogonal and nonorthogonal grids.

Table 2. Reattachment length as a function of Reynolds number for backward-facing step problem

Scheme	Grid	Re=133	Re=267	Re=400	Re=600
Thompson and Ferziger	256 × 64	4.0	6.5	8.5	10.1
Thompson and Ferziger	512 × 128	-	-	8.7	10.8
OCV	121 × 61	4.0	6.6	8.7	10.7
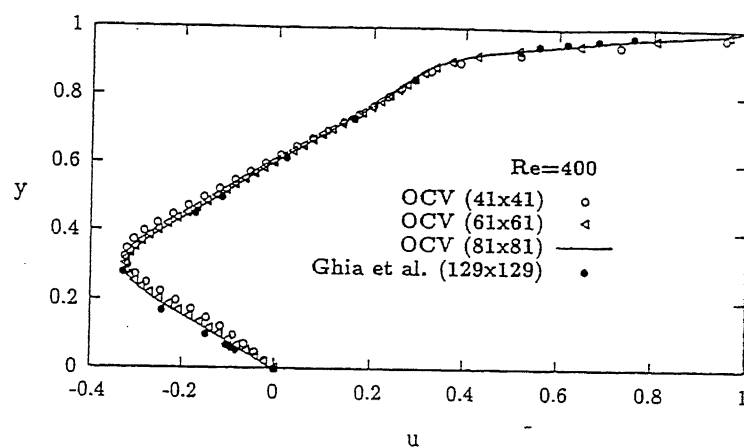
Velocity profiles on vertical centerline of a square cavity, Re=400
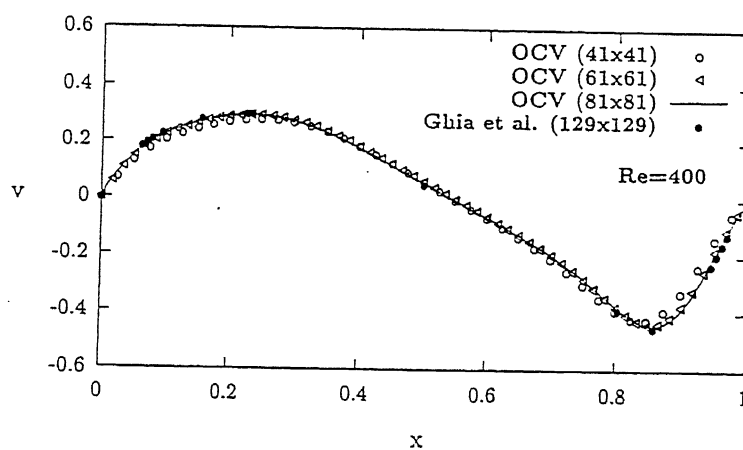
Velocity profiles on horizontal centerline of a square cavity,
Re=400

Fig. 3. Results for driven square cavity problem (Re=400)

Contents

List of Figures

List of Tables

Nomenclature

A, A_s	Area of a control volume
D	Diameter
H	Channel height
J	Jacobian
L	Length
N	Number of grid points
N_i	Shape functions for velocity compoents, pressure or any scalar functions
n_x, n_y	Direction cosines of outward unit normal on control volume faces
p	Pressure
R_e	Reynolds number
S_p	Source term
t	Time
u	velocity component along x-direction
U	Dimensionless axial velocity, u/U_{av}
u	velocity vector
v	velocity component along y-direction
V	volume
x, y	Cartesian coordinates
x_1, y_1	Cartesian coordinates

Greek letters

ξ, η	body-fitted coordinates or coordinates in the transformed space

μ dynamic viscosity
ν kinematic viscosity
ϕ scalar component
ψ streamfunction, pseudo pressure
ω relaxation factor
Ω solid angle
ρ density

Subscripts

av	average
e, w, n, s	surface points on faces of the control volume
E, W, N, S	compass directions
n	time level
n	iteration index
x, y, z	in a given direction

Superscripts

(1), (2), (3), (4)   face number of a control volume

Chapter 1

Introduction

The prediction of incompressible fluid flow is needed in almost every branch of engineering. Examples include chemical processing plants, nuclear reactors, heating and ventilation of rooms, cooling of electronic equipments, meteorology, turbomachines, groundwater flows, etc. As the field of computational fluid dynamics continues to mature, there is a need to develop new algorithms which are capable of exploiting the most recent advances in numerical mathematics, computing architectures and hardware to solve fluid flow problems of increasing complexity, for industry and in fundamental research.

Successfully obtaining physically meaningful solutions from numerical flow predictions depends on (a) the correctness of the mathematical approximations in the modelling of the physical systems, and (b) the accuracy and efficiency of the numerical scheme and its ability to handle the complexity of the flow geometries under consideration. In this thesis we are primarily concerned with the second of these two factors and present a calculation procedure, for incompressible 2-D flows in complex geometries, that was developed with the aim that it be accurate, efficient and robust.

The goal of the laminar, incompressible flow simulation is usually to solve the Navier-Stokes and continuity equations, which give a very good description of the velocity and pressure fields of a laminar newtonian fluid flow. If the numerical solutions are obtained using reliable schemes, the solutions generally have good accordance with

experimental values.

The situation is more uncertain for turbulent flows, since in the Reynolds-averaged Navier-Stokes equations there is uncertainty associated with the semi-empirical turbulence models that need to be used to close the system of equations, and no guarantee exists that the computed flow properties will compare well with experimental data, in particular for sensitive variables such as wall shear stresses and heat transfer coefficients. Although empirical turbulence modelling is not required for Direct Numerical Simulations of turbulence, the computational resources required for flow simulations of practical relevance are exceedingly high and will probably be outside the reach of the typical researcher for many years. Therefore it is likely that turbulence models, despite all their uncertainties will be in use for some time. However apart from the physical validity of the turbulence models, most common models like the $k - \epsilon$ model do not pose fundamental *numerical* difficulties, and a laminar flow solver can relatively easily be adapted to accommodate these models. Thus it is usual that new schemes for CFD, when developed, are created to solve laminar flow, and later adapted for turbulent flows if necessary.

Viscous incompressible fluid flow can also be simulated by using the vorticity-stream function formulation. This formulation has certain advantages in that pressure is eliminated from the governing equations resulting in a substantial simplification of the solution algorithm (for obtaining the pressure is a major task in Navier–Stokes solvers). This approach also reduces the number of equations that need to be solved. However the vorticity stream-function formulation suffers from the major drawback that is applicable only in two-dimensional cases, or for the special case of three dimensional axisymmetric flows. Thus general-purpose viscous flow solvers are usually based on the Navier-Stokes equations, even if they are initially developed, as here, for two-dimensional flow. The formulation based on Navier-Stokes equations is called the *primitive variable* formulation, with the components of velocities and pressure being the so-called primitive variables.

The two most important factors requiring consideration in numerical schemes for in-

compressible Navier-Stokes solvers are (a) the discretization of (nonlinear) convection terms, and (b) the pressure field calculation :

(a) The diffusion terms can be discretized by the second-order accurate center-difference scheme and thus usually do not pose a problem. However, center differences tend to be numerically unstable for the convective terms which have traditionally been discretized by a first-order upwind scheme[1]. However, this upwind scheme introduces substantial error in the form of *false diffusion*, i.e. smearing of the numerical solutions, and has been in recent years yielded in popularity to other schemes of greater accuracy, e.g. hybrid schemes[2,3,4], second-order upwind differencing schemes[5], QUICK[6] and its variants, etc., which usually incorporate some sort of upwinding for reasons of numerical stability as well as to obtain the physically correct transportive property. Another problem that is sought to be addressed by the convection discretization is that of *cross-stream* diffusion, i.e., errors which appear when the flow is not aligned to the grid lines. As CFD is used for increasingly complex flows, the need to minimize cross-stream diffusion is paramount. Yet another problem that needs to be addressed is caused by the increased accuracy of modern schemes – second-order schemes for convection often display *dispersion errors*, i.e., unphysical over-/under-shoots in the solution in the presence of strong gradients. These errors are sought to be contained by *flux limiting* (see below).

Given the many aspects of accuracy, stability, boundedness, etc., involved in convective term discretizations, new schemes for convection modelling are tested not on the cumbersome Navier-Stokes equations, but on the convection-diffusion equation which also has all these problems of convective and diffusive term discretization. This simpler equation allows us to focus directly on the problems of discretization and delink them from the other problems associated with the Navier-Stokes equations. In this thesis, we therefore test our discretization schemes extensively on the convection-diffusion equation before applying it on the Navier-Stokes equations.

(b) Another problem which is mainly relevant to the Navier-Stokes equation discretization is that of *pressure-velocity decoupling*[4]. This problem is rooted in the fact that the *incompressible* Navier-Stokes equation has no separate equation for pressure but instead has a compatibility condition, the continuity equation, for the components of

velocity. Thus the pressure field is *correct* if it yields a divergence-less (i.e., continuity satisfying) flow field. In fact any *pseudo-pressure* field which does this is acceptable from the view-point of velocity field correctness. (The *true* pressure is a field which fulfils the above condition *and* also satisfies the pressure boundary conditions). Because the acceptability of the pressure field depends on compatibility conditions on velocity, both quantities have to be solved simultaneously, in incompressible flows, by iterating the pressure and velocity fields. If the same grid points are used for pressure and velocity, and no preventive measures are taken, these iterations do not converge because the pressure and velocity fields decouple, i.e., changes in one do not necessarily communicate to the other. This can easily and naturally be prevented by using a staggered grid[4], in which pressure and velocity grid points have different locations. However if non-staggered grids are used, for there are compelling reasons to use such grids in complex domains (see below), methods have to be found to prevent the above mentioned decoupling.

The choice of an appropriate numerical method often depends upon the geometry of the flow domain. The most commonly used solution methodologies are finite-difference, finite-volume and finite element methods. Both finite-volume and finite-element methods belong to the class of weighted residual methods[7]. However, in implementation, finite volume discretizations more closely resemble finite difference methods. Finite-difference methods[7], in general, are algorithmically simplest, while finite-elements[8] are the most flexible with regards to flow geometry, but are rather involved algorithmically. The finite volume method is often seen as a good compromise between the two – being simpler to implement than the finite-element method and being more suitable for irregular domains than the finite difference method. The finite volume method has the additional advantages of discretizing, directly, the conservation form of the governing equations. This implies that the discretised equations preserve the conservation laws. This is a particular advantage when obtaining accurate solutions for internal flows or flows with shocks.

If the geometry is rectangular then finite-difference methods are often preferred for their simplicity. However, with the growing use of CFD in industrial applications, compu-

tations are done on increasingly complex flow geometries. The recent advances in grid generation techniques[9,10,11] allow the use of boundary-fitted grids which permit selective refinement near the boundary walls, etc., and thus reduce storage, computational time, and improve accuracy. However finite-difference schemes used on body-fitted coordinates require us to map the complex domain onto a regular rectangular grid on which transformed equations are solved. This could be global mapping, i.e., mapping the entire flow domain onto a simpler computational domain, or, more commonly, local mapping, i.e., mapping each successive grid point and its relevant neighbours onto a succession of small and simple domains. The transformation approach is often adopted even for finite-volume schemes. The transformed equations, however, are quite unwielding if the grid is not orthogonal[7]. For transformation disguises a number of difficulties by the apparent simplicity of the transformed (computational) domain. These problems are :

- The complexity of the original geometry is stored in the transformation parameters which are incorporated into the transformed governing equations, which are therefore correspondingly more complicated and cause significant increase in computational effort.

- The unevenness and contortions of the physical domain translate to abrupt changes in the transformation parameters and therefore effect the accuracy even on the seemingly smooth domain.

- The choice of velocity components obtained in the transformed system can be (a) contravariant (b) covariant or (c) the usual Cartesian/cylindrical components. These have their relative advantages and disadvantages, the first two cause the solutions to be sensitive to grid-dependent curvature terms, and the last requires extensive interpolation.

Given these difficulties with methods based on transformation of the governing equations, other methods have been evolved which solve the governing equations in the physical domain itself where the governing equations are much simpler than their counterpart in the transformed system. These are the *true* finite-volume methods, which do not seek an escape from complex geometry by transformation into a simpler one

but solve the governing equations directly on the complex domain, usually on a non-orthogonal grid. Furthermore, Cartesian components are the natural variables in this approach, and do not introduce additional complexity into the equations. However, this approach requires us to have a more flexible geometric representation of each finite volume, and to develop more flexible ways to discretize the convective and diffusive terms, keeping in view the considerations mentioned above.

The most commonly used grids for finite-difference and finite-volume methods are either staggered or non-staggered (collocated) grids. The staggered grid arrangement is quite robust and efficient in orthogonal grid computations and avoids pressure-velocity decoupling associated with non-staggered grids. On uniform grids, the grid points have a simple relationship with the grid index and their locations need not be stored. However, on non-uniform grids, the coordinates of each grid point need to be stored. On such grids, a staggered system will store many times more information than non-staggered ones, as each component of velocity and the pressure have different locations in the former. In addition, the grid transformation parameters for each variable is different on staggered grids, and again need to be stored. This also poses problems in the discretization of the non-linear terms on a non-uniform mesh. All this is heavy price to pay for the staggered grid's pre-eminent advantage — the avoidance of pressure-velocity decoupling.

On the other hand, the collocated grid arrangement stores all the dependent variables at the same grid points and only one set of grid relations needs to be considered. This arrangement is susceptible to pressure-velocity decoupling, but this can be avoided by using momentum interpolation[12]. In view of the above considerations, the collocated grid arrangement seems preferable for fluid flow calculation in complex geometries.

In this thesis, we attempt to develop a finite volume scheme applicable directly on structured non-orthogonal grids in arbitrary geometries. By using an isoparametric formulation, general quadilateral elements can be introduced to represent complicated computational domains. A collocated grid arrangement is chosen for the storage of

the dependent variables to facilitate calculation in complex domains. The problem of pressure-velocity decoupling is avoided using momentum interpolation. The Cartesian components of velocity and pressure are used as dependent variables.

1.1 Discretization Schemes for Convection

A major difficulty in the numerical modelling of flow equations is in discretizing the convective term. A good scheme should possess the following properties : accuracy, stability, boundedness and algorithmic simplicity. These requirements are often in opposition with one another. Stability and boundedness require that the scheme have diffusive smoothing, whereas accuracy suffers by this.

The use of central-differencing for the convective term makes the numerical scheme unstable for grid Peclet numbers greater than 2.0. This numerical instability can be avoided by using first-order upwind differencing. First-order upwind schemes can generate significant errors, the most serious of which is due to a diffusive effect which mimics the effects of viscosity. The interaction of this *artificial diffusion* and the real diffusion arising from viscosity was considered by Roache[13]. The false diffusion associated with the first-order upwind scheme can cause the numerical solution to severely misrepresent the physical transport process[14]. Thus in CFD, false diffusion is a major obstacle to the achievement of accurate solutions to the governing equations[15].

To improve on the accuracy of first-order upwind schemes, one can switch to more accurate central differencing for convection or use a suitably blending of the central and simple upwind approximations[2,3,4]. These schemes ensure conditional stability and boundedness, and the results are quite acceptable for problems with essentially a unidirectional velocity aligned with the grid line. However, in applications on more complex problems involving convection and any combination of streamwise diffusion, cross-stream transport in multidimensions, and source terms, etc., these schemes are generally not accurate and introduce considerable false diffusion[6].

In addition to the errors due to streamline curvature there is also the problem of streamline-to-grid skewness in multidimensions[15,16]. The skew upstream differencing

scheme[17] (SUDS), QUICK[6], and its variants[18,19,20] are able to reduce the level of false diffusion significantly for oblique flows. SUDS is a first-order upwinding applied along the skewed streamline passing through the cell-face of interest. In QUICK, the convective flux is estimated from a second-order polynomial fit between two upstream and one downstream nodes. The second-order upwind differencing[5] has been suggested as an alternate to first-order upwinding. In this scheme, the convective flux is estimated from two upstream nodes. It has been shown in various studies[21,22] that the second-order upwind scheme has better performance than the first-order upwind or hybrid schemes.

Higher-order schemes are able to reduce the level of numerical diffusion significantly. However, they produce unphysical oscillations in regions of strong gradients. Just as numerical dissipation dominates error in first-order schemes, dispersion errors, which cause oscillations, may be significant in second-order ones. The one of the motivations behind the development of upwind schemes was the hope that the generation of numerical oscillations associated with the space centered schemes can be removed by introducing propagation properties in the discretization. However, the straight-forward replacement of the first-order upwind schemes by higher-order schemes leads again to the generation of oscillations around discontinuities similar to that in space centered schemes. It has been shown theoretically by Engquist and Osher[23] that linear second-order upwind schemes always generate oscillations.

We discuss below schemes that have been used to reduce oscillations in the solution of a *single scalar* convective equation, though many of these schemes have been used in other circumstances.

Godunov[24] introduced the important concept of monotonicity. The *monotonic schemes* retain the monotonicity of an initially monotone solution of the convective (wave) equations. Godunov showed that all monotone linear schemes can be at most of first-order accuracy. For non-linear equations, the concept of a bounded total variation of the solution is more general and has been introduced by Harten as a criterion to check that unwanted oscillations are not generated by a numerical scheme. The *Total Variation Diminishing* (TVD) schemes of Harten[25] are designed to ensure that total

variation, *i.e.*, the sum of the magnitude of differences in the solution between adjacent grid points, always decreases in the solution. It was shown that these schemes can only be first-order accurate.

The only way to obtain non-oscillatory second-order schemes is to introduce non-linear discretization, i.e., schemes which will be non-linear even when applied to linear equations. This important concept was initially introduced by Van Leer[26], and Boris and Book[27], in the form of *limiters* which control the gradients of the computed solution so as to prevent the appearance of over- or under-shoots. The Flux-Corrected Transport (FCT) scheme of Boris and Book[27] was modified and enhanced by Zalesak[28]. Some other schemes in this class are the ENO (Essentially Non-Oscillatory) schemes of Shu and Osher[29], the slope-modification method of Yang[30], and the TVD scheme of Wang and Widhopf[31]. Hirsch[32] provides extensive reviews of TVD and other high-order schemes.

The TVD diagrams of Sweby[33] and the Normalized Variable Formulation (NVF) methodology of Leonard[34] provide a conceptual framework for the development and analysis of high-resolution convection-diffusion schemes. The SMART[35], SHARP[36] and UMIST[37] schemes are monotonic implementations of Leonard's third-order QUICK[6] scheme. These formulations switch between QUICK and lower-order schemes depending upon the local value of a parameter used to identify the presence of an extremum. Darwish and Moukalled[38] developed a new Normalized Variable and Space Formulation (NVSF), in which spatial parameters are introduced so as to extend the applicability of NVF methodology to non-uniformly discretized orthogonal domains. However, *most flux limiters have been developed for schemes for uniform meshes or at most nonuniform orthogonal meshes.* These can only be used for non-orthogonal meshes if the grid is transformed onto an orthogonal one.

The other possible approaches for reducing over- and under-shoots involves a local blending of a higher-order method and a weighted contribution from lower-order method[39] (flux blending) or by introducing a controlled amount of numerical diffusion based on local gradients[40].

The basis of evaluating flux limiting schemes are generally (a) boundedness : that the

scheme eliminates spurious oscillations, (b) minimal diffusion : that it not *spread* the solution by introducing too much diffusion, (c) accuracy : that it not cause too great a degradation of accuracy, (d) non-compressive : that it not tend to *square-off* a smooth solution. Most schemes that have been developed to eliminate oscillations in practical multidimensional flow equations are tested on the convection-diffusion equation.

1.2   Finite-Difference and Finite-Volume Methods for Regular Geometry

In this section, we discuss the most widely used numerical schemes for the solution of the Navier-Stokes equations in regular (simple) geometries.

The marker and cell (MAC) method[41] is the oldest of the practicable incompressible Navier-Stokes solution schemes. Though developed initially for free-surface flow simulations, the MAC method can be used for general incompressible Navier-Stokes problems. In the original implementation, the velocity is advanced by an explicit time integration and a modified pressure equation is designed to ensure an approximate enforcement of continuity at the next velocity update. The MAC method is implemented on a staggered mesh in which the pressure is computed at cell center, and the velocities are computed at cell faces. Several variants of the MAC method exist[42,43,44], most of which enforce continuity at the *current* time-step. Abdallah[43] proposed a modified MAC method based on nonstaggered grids.

The most widely used variant of MAC is the SMAC[42] (Simplified Marker and Cell method) which uses a two step predictor-corrector procedure : the first step is fully explicit and the second step introduces corrections to velocity and pressure to satisfy continuity. These corrections are obtained by solving a pressure-correction Poisson equation. The use of this Poisson equation, is the primary distinction between MAC and SMAC. Several variants of SMAC too exist[45]. In SMAC, the true pressure is usually not calculated. Since the stream-function vorticity formulation does not contain the pressure, it is clear that pressure is not fundamental to obtains the velocity field. Thus *any* pressure field inserted into the Navier-Stokes equations which enforces continuity

will give the correct velocity field. (It is important that the velocity field should also be mass conserving at the wall). Assuming that the velocities from the previous time step are divergence-free, then if a pseudo-pressure is computed such that the new velocities are mass-conserving, then the new velocities will be the unique and correct velocity field. The advantage of using a pseudo-pressure rather than the true pressure is that homogeneous Neumann boundary conditions can be used with the former, while using the actual boundary conditions for pressure is difficult as there are *no* boundary conditions that can be *a-priori* specified for the pressure – the boundary conditions have to be obtained from the Navier-Stokes equations.

Chorin[46,47] introduced the concept of an artificial compressibility factor into the equations of motion and obtained incompressible flow solutions as a limiting case of pseudo-compressible flow. This formulation is based on the primitive variable approach and uses a nonstaggered grid arrangement. The modified momentum equations are integrated using an implicit ADI scheme. In the finite-difference applications, it is now implemented with an explicit time integration and a staggered mesh.

The Semi Implicit Method for Pressure Linked Equations (SIMPLE) of Patankar and Spalding[48] is a widely used algorithm. It uses a finite-volume method for discretizing the incompressible Navier-Stokes equations on a staggered grid. The SIMPLE algorithm uses the momentum and continuity equations to link pressure and velocity corrections and iterates till the velocity field is divergence-free. No-slip and free-slip boundary conditions are imposed using the reflection method for the velocity, and a homogeneous Neumann condition is used for the pressure. Variants of SIMPLE differ mainly in the ways the pressure and velocity corrections are computed. Connell and Slow[49] compared the convergence rates of the SIMPLE algorithm to the results obtained using two extended pressure-correction equations which included advection terms, dropped in the SIMPLE formulation. Wen and Ingham[50] proposed a new pressure correction formula for a SIMPLE-like algorithm in order to improve the rate of convergence when solving laminar flows with rapidly varying pressure. A modified algorithm called SIMPLE Revised[4] (SIMPLER) was developed to improve the convergence rate of the pressure field solution. The SIMPLEC algorithm[51] is another variation of the basic SIMPLE algorithm. SIMPLEC eliminates the approximations made in SIMPLE while deriving the pressure correction equation and removes the need for an under-

relaxation parameter to update pressure. In both SIMPLE and SIMPLEC procedures, the pressure correction is used to update the pressure, and correct the velocities. That SIMPLEC and SIMPLER are substantially more economical than SIMPLE, and that SIMPLEC is usually less expensive than SIMPLER was reported by Vandoormal and Raithby[51] in their study for two recirculating flow problems.

The Pressure-Implicit with Splitting of Operators (PISO) method[52] is a noniterative predictor-corrector algorithm used for incompressible and low-Mach number compressible CFD applications. It uses a finite-volume staggered grid arrangement and time integration is carried out using a fully implicit (backward Euler) scheme. Jang, Jetly and Acharya[53] presented PISO in SIMPLE-like notation for easy comparison with the SIMPLE family of algorithms. They found that for isothermal laminar flows, where the momentum equations are not strongly coupled to a scalar transport equation, the non-iterative PISO algorithm outperformed the iterative SIMPLER and SIMPLEC method. However, for problems with a strong coupling between the momentum and scalar transport equations (e.g. coupling with the energy equation through a Boussinesq buoyancy term, or with a $k - \epsilon$ turbulence model through the eddy viscosity), SIMPLE and SIMPLEC did better. PISO also required small time steps for the strongly coupled cases in order to obtain acceptable solutions.

The relative performance of the staggered grid method and non-staggered method have been investigated[54,55] in regular geometries. Peric, Kessler and Scheuerer[54] demonstrated that the convergence rate, dependency on under-relaxation parameters, computational effort and accuracy are almost identical for both solution methods. They found that collocated grid method converges faster in some cases, and has advantages over staggered grid method when extensions such as multigrid techniques and non-orthogonal grids are considered. Kobayashi and Pereira[56] presented a revised version of SIMPLE for flow calculations using nonstaggered, non-orthogonal grids. They used Cartesian components of velocities and the pressure-velocity decoupling problem was avoided by using the modified momentum interpolation of Majumdar[57]. Date[58] showed that the momentum-compatible cell face velocity differ from the arithmetic mean velocity by an amount that reflects the non-linearity in the local pressure. A method for prediction of pressure using an effective pressure gradient in the nodal momentum equations while still evaluating cell face velocities as arithmetic mean velocities was also

proposed. However, this method was found to yield oscillating pressure distribution in situations where pressure variations are steep. Date[59] presented a new derivation of the pressure-correction equation appropriate for a nonstaggered grid. It consists of two parts: a mass conserving component and a smoothing component. The former corresponds to the pressure correction predicted by a staggered grid procedure, whereas the latter simply accounts for the difference between the point value of pressure and the cell-averaged value of the pressure.

Aksoy and Chen[60] presented a modified SIMPLEC method for nonstaggered grids. The momentum-weighted interpolation method (MWIM) calculates velocities and pressures at cell centers. MWIM is very similar to another variant of SIMPLE called momentum interpolation[12]. To avoid spurious pressure solutions, the divergence error is evaluated using velocities interpolated to cell faces. A comparative assessment of the performance of MWIM, SIMPLER, SIMPLEC, PWIM and a nonstaggered grid version of MAC, was carried by Abdallah[43], for the lid-driven cavity problem. They found deterioration in the performance of all the algorithms as the grid size was refined. The nonstaggered version of MAC had excellent convergence and stability properties as an explicit method; however, conservation of mass was poor near flow singularities. MWIM and PWIM produced more or less the same performance characteristics.

1.3 Finite-Difference and Finite-Volume Methods for Complex Geometry

Several methods have been proposed to solve the incompressible Navier-Stokes equations in arbitrary geometries in conjunction with orthogonal and nonorthogonal grid systems. These methods differ in the choice of grid layout (staggered versus nonstaggered) and dependent variables in the momentum equations (Cartesian versus curvilinear velocity components).

The dependent variables in the momentum equations are velocity components. For simple coordinate systems, such as Cartesian and cylindrical, the choice of the dependent variables is obvious. For a non-orthogonal coordinate system and a transformed

system of equations the selection is not straightforward. One can choose contravariant or covariant components of velocity or Cartesian/cylindrical components. Covariant components are defined along the grid lines whereas the contravariant components of velocity are oriented normal to the control-volume surface in the generalized coordinate system[61]. Therefore on an orthogonal grid, covariant and contravariant components are identical. Since the contravariant components are normal to the faces, the calculation of mass fluxes is straightforward with this system but the stencil involved in the Poisson equation for pressure requires nine grid points (in 2-D) and lacks diagonal dominance. On the other hand, the use of covariant velocities results in a compact (5 grid point) stencil for the pressure Poisson equation. However, the calculation of mass-fluxes is not straightforward and requires all components of velocities at each face, and on staggered grids only one component is available and the others need to be obtained by interpolation from the neighbouring points.

Contravariant and covariant components of velocity change their direction and tend to follow the grid lines. This feature makes them useful for non-orthogonal grids. But, due to the change in their direction, the governing equations are very complicated and involve sensitive *curvature* source terms. The Cartesian components of velocity are used as a dependent variables in non-orthogonal coordinate systems and the resulting governing equations are much simpler than their counterparts for the curvilinear velocity components. However, neither the mass flux nor the pressure is conveniently represented in most Cartesian formulations, as the components are neither aligned with the face normals nor with the grid lines. This usually means that extensive interpolation is needed to compute components at the cell faces and this interpolation often causes loss of accuracy.

The use of an orthogonal coordinate system is preferable since the equations are simple and calculation procedures developed for a standard coordinate system, such as Cartesian, can be easily modified for orthogonal coordinates. However, such grids are difficult to generate for irregular geometries and very little control is left over the grid spacing. For nonorthogonally intersecting boundaries, strictly orthogonal coordinates can not be obtained. The grid arrangement used with orthogonal coordinate systems usually has been staggered[62,63,64,65,66].

The choice of nonorthogonal curvilinear grid provides more flexibility in the choice of grid points and one can refine the grid locally in regions where this is necessary to maintain solution accuracy. With curvilinear coordinates, it is possible to map the grid lines from the physical domain to a simpler computational domain. Either the differential form or the integral form of the governing equations are then transformed and solved on the computational domain. In finite-difference methods the differential governing equations are used while in the finite-volume methods the integral forms are solved. In either case, the transformed equations are quite complicated and contain grid-sensitive parameters.

In the primitive variable formulation it has been the tradition to use staggered grids to circumvent the problem of pressure-velocity decoupling. It is possible to use any of the velocities (Cartesian, covariant and contravariant) as dependent variables on a staggered grid. Cartesian components as dependent variables are not very effective with a staggered grid when the grid is highly non-orthogonal and results in poor coupling between velocity and pressure. As an alternative, both Cartesian velocity components can be stored at the each control-volume face[67], but this arrangement will require extra computational effort even in the regions where the grid is orthogonal or nearly orthogonal. The curvilinear velocity components, covariant or contravariant, can be expected to give better coupling between velocity and pressure fields. To derive equations for covariant or contravariant velocities, one can deduce governing equations for grid-oriented velocities by using tensor calculus[68,69,70,71]. The transformed governing equations are complex due to the presence of extra source terms arising from the spatial variation of the base vectors. Another option is to use a locally fixed coordinate system[61]. In this approach, discretized equations based on Cartesian velocity components are modified to give equations for the grid-oriented velocity. The locally fixed coordinate system seems to be better than the global curvilinear coordinate system[72]. The coupling between continuity and momentum equation is usually effected using SIMPLE-type algorithms[61] or SMAC procedure[73,74,75].

Since there are three related grid systems (for u, v, and p) in the staggered grid system, the transformed form of each governing equation, apart from being quite complex in a curvilinear grid system and requiring more computational effort, also require the storage of a complete set of grid parameters for each dependent variable. This is because

the grid points of the variables are not identically located. It is also not straightforward to incorporate multigrid methods with staggered grid arrangement in complex geometries. It is thus simpler to work with nonstaggered grid arrangement in generalized coordinate system, compared to a staggered one, but the pressure-velocity decoupling problem has to be avoided. Also, in incompressible flow calculations, since Cartesian velocity components are the simplest they are the preferred dependent variables in a nonstaggered grid arrangement.

Hsu[76] and Rhie and Chow[12] successfully used the nonstaggered arrangement for numerical calculation of incompressible flow with curved boundaries employing the SIMPLE algorithm. The pressure-velocity decoupling was avoided in the Rhie and Chow[12] formulation by using *momentum* interpolation to compute cell face velocities. Majumdar[57] proposed an implementation of momentum interpolation to achieve solutions independent of the underrelaxation parameter used.

Reggio and Camarero[77] presented a calculation procedure for solving the time-dependent incompressible Navier-Stokes equations in arbitrary geometry using collocated grids. Their numerical scheme is based on an overlapping grid with forward and backward differencing for mass and pressure gradients, respectively, to avoid pressure-velocity decoupling and the governing equations are formulated in generalized curvilinear coordinates. The momentum equations involve curvilinear and Cartesian velocity components, whereas mass conservation depends only on the contravariant velocity components.

Melaaen[72] compared methods for staggered and nonstaggered grids used for calculating flows in curvilinear nonorthogonal coordinates. Covariant velocity projections are used for the staggered arrangement and Cartesian velocity components with the nonstaggered grid arrangement. He found that computations using a very skew and non-uniform grid was best calculated by the nonstaggered method; while in flows with high pressure–gradients, the pressure field calculated by the staggered method seems to give grid-independent solution on a coarser grid than the nonstaggered method.

Cheng and Armfield[44] developed a SMAC time-advancing method for solving the unsteady incompressible Navier-Stokes equations on non-staggered grids in generalized

coordinate systems. They used Cartesian velocities and pressure as the dependent variables. A special flux correction at the faces of the finite volume was utilized in the continuity equation to suppress pressure oscillations. The comparisons show that this SMAC method is more efficient than SIMPLEC and PISO methods for the steady and unsteady flows considered.

Armfield[78] described an elliptic flux correction method to prevent the grid-scale pressure oscillations. The scheme is similar to that of Rhie and Chow[12] when applied on an unsteady problem with a rectangular coordinate system, but without the relaxation error identified by Majumdar[57]. It was also demonstrated that schemes of this type effectively introduce an additional discrete term into the continuity equation which leads to the nonstaggered scheme having an identical ellipticity to the standard SIM-PLE scheme. Without the additional term, nonstaggered schemes are non-elliptic at the grid-scale wave number, resulting in the pressure oscillations.

Lee and Chin[79] developed a covariant velocity based procedure for a nonstaggered grid arrangement. The interpolation of the covariant velocity components is done indirectly by interpolating the contravariant velocities. They found that the direct interpolation of covariant velocity components creates virtual velocities that lead to an erroneous solution.

The methods mentioned above for collocated or non-staggered grids are meant for curvilinear coordinate systems and are meant to be used with transformed equations, and thus have all the associated problems. Peric[80] generalized the idea of Rhie and Chow[12] to solve 2- and 3-dimensional flows in the physical domain itself. The use of Cartesian velocity components retains the so-called strong-conservation form of governing equations and totally eliminates the need to calculate the extremely grid-sensitive curvature terms[80]. Rodi et al[81] and Majumdar et al.[82] have developed the flow solver for general 2- and 3-dimensional flows, respectively and were inspired by the work of Rhie and Chow[12] and Peric[80]. Demirdzic, Lilek and Peric[83] extended the 2-dimensional method for prediction of steady-state incompressible flows in complex geometry to also treat compressible flows at all speeds.

1.3.1   Other Methods

Some other methods have been developed for the solution of fluid flow equations in complex geometries. The control-volume based finite-element method retains the geometrical flexibility of the finite-element method and derives the governing discrete algebraic equations by using a conservation balance applied to discrete control-volumes. Baliga and Patankar[84] introduced the formulation of a control volume finite-element method (CVFEM) for the solution of two-dimensional incompressible fluid flow and heat transfer problems in arbitrary shaped domains. The method employs triangular elements to discretize the physical domain and uses shape function which is exponential in the direction of the average velocity vector and linear in the normal direction. This choice of shape function was made to reduce false-diffusion. As in may other finite-element methods for the solution of fluid flow equations, a shortcoming of this method[84,85] is that it uses an unequal order interpolation of velocity and pressure. Thus, whereas velocity components are computed at all grid points in the domain, pressure is computed at much fewer grid points. This has made their method quite complex. This shortcoming was later alleviated by incorporating equal-order interpolation for both velocity and pressure[86,87]. All the above CVFEM methods use an iterative procedure akin to SIMPLER[4]. The application of CVFEM to the 3-dimensional convection-diffusion equation[88] and the relative performance of unequal and equal-order CVFEM has been extensively studied[89]. Some other types of upwind scheme like mass-weighted upwind[90] scheme have been incorporated with CVFEM[91]. Masson et al[92] used CVFEM to solve 2-dimensional axi-symmetric incompressible flow problems. The collocated CVFEM was extended to quadrilateral elements by Schneider and Raw[90] in a manner designed for accurate convection modelling and preclusion of pressure field decoupling.

Mukhopadhayay, Biswas and Sundarajan[93] presented an 2-dimensional algorithm for non-orthogonal geometries using primitive variables. Ideas such as elementwise interpolation and mapping of non-orthogonal elements into a square transformed domain, commonly used in FEM, have been incorporated and pressure nodes are staggered with respect to velocity nodes to avoid pressure-velocity decoupling. Many of these ideas have been inspired the current work which however is for a non-staggered arrangement.

Jessee and Fiveland[94] presented a cell vertex algorithm for the incompressible Navier-

Stokes equations, on non-orthogonal grids for 2- and 3- dimensional geometries, for Cartesian components of velocity. Advective fluxes were calculated by bounded, high resolution schemes used with a deferred correction procedure to maintain a compact stencil. The mass and momentum equations were solved a nonstaggered grid. The coupling between the pressure and velocity was achieved by using the Rhie and Chow[12] interpolation scheme modified to provide solutions independent of time steps or relaxation factors.

1.4   The Scope, Objectives, and Organization of the Thesis

The objective here was to develop a solver for two dimensional flow for complex geometries and to thoroughly test and establish its robustness, accuracy and computational efficiency in a variety of flow problems.

The solver was developed with the following points in mind :

- The governing equations be solved directly on the physical domain (rather than a transformed domain) in view of simplicity of the equations in the former.

- The method should be finite-volume, as this method has shown itself to often deliver a good compromise between geometric flexibility, accuracy, and algorithmic simplicity. The grid should be non-staggered (collocated) as this is more natural for non-orthogonal grids.

- A proper means of higher-order upwinding for the convection terms be implemented to ensure accuracy and stability.

- Convective and diffusive terms be computed with second-order accuracy, at least on uniform grids.

- A flux limiter be developed to suppress the spurious oscillations that are known to develop in second-order upwind schemes.

- The solver be tested on steady-state as well as on transient convection-diffusion problems; and for transient problems, both explicit and implicit time-stepping be implemented.

- The solver should retain its accuracy over a large range of grid-Peclet numbers. For explicit schemes it should be numerically stable for Courant numbers upto unity, and in implicit schemes for Courant numbers significantly greater than unity.

- The method should be applied to the Navier-Stokes equations. A method to overcome the difficulty of pressure-velocity decoupling, common on non-staggered grids, should be found.

- Each of the above features should be credibly demonstrated by through validation on test problems.

The goals set out above have substantially been met. The basic finite-volume scheme developed in this thesis is called OCV (after Over lapping Control Volume) and its properties are extensively tested on the convection-diffusion equation and it is then applied to the Navier-Stokes equations. The OCV method uses an iso-parametric formulation to compute diffusion and to introduce higher order upwinding. However, it avoids the need for assembly, common in finite-element algorithms.

The details of the discretization schemes used for the diffusion and convection terms in the OCV method and their order of accuracy are presented in chapter 2. It is demonstrated there that the accuracy of the discretization schemes used for the diffusion as well as convection terms are second-order on uniform grids. The capabilities of the discretization procedure to handle non-orthogonal grids is demonstrated. It is observed that the solution accuracy of the OCV method is not much degraded by mild distortion in the grid.

It is well known that second-order schemes show unphysical spurious oscillations in the solution for convection dominated flows with steep gradients. In some cases this may even cause numerical calculations to diverge. In chapter 3, we present a flux limiter specially developed for this formulation. The efficacy of this flux limiter is demonstrated by solving a variety of steady-state convection diffusion problems. The

flux limiter is found to be effective with explicit time-stepping, and is quite successful in removing oscillations, without introducing too much numerical diffusion, and thereby approximately retaining the accuracy of the base discretization scheme. In chapter 4, implicit time-stepping schemes for the OCV method are used to solve the transient advection-dispersion equation. The effect of grid-Peclet number and Courant number on the solution accuracy and stability of the scheme are studied in detail and it is demonstrated that it can be used for large range of grid Peclet numbers and Courant numbers, and is suitable for both diffusion and convection dominated flows. The flux limiter is effective in controlling spurious oscillations for Courant numbers below unity. For higher Courant numbers, numerical dissipation needs to be introduced to suppress the oscillations, but the flux limiter helps in improving the numerical stability of the time-stepping.

An OCV algorithm for computing steady and unsteady solutions for the two-dimensional incompressible Navier-Stokes equations on nonstaggered grids is presented in chapter 5. The method uses equal-order interpolation scheme for the dependent variables (u, v and p). The problem of checkerboard pressure distribution has been avoided by using an idea similar to Rhie and Chow[12]. The Navier-Stokes solution algorithm involves several considerations, such as the differencing of pressure gradient term, calculation of face-center velocities and the fluxes across the faces of the control volumes. This formulation for the OCV method can be used on non-orthogonal grids without any modification. It is demonstrated there that the method performs well on a variety of test problems.

A part of the work presented in this thesis has been already published[95], accepted for publication[96,97], or has been communicated for publication[98,99].

Chapter 2

Overlapping Control Volume Approach for Convection–Diffusion Problems

2.1 Introduction

In fluid flow and heat transfer problems, convection and diffusion are usually of primary interest. Therefore a numerical scheme for fluid flow and heat transfer can be tested on the convection-diffusion equation which, while incorporating similar processes, is simpler to solve, and, being linear, has some known analytic solutions.

In this chapter, we develop a scheme, applicable directly on non-orthogonal grids in arbitrary geometries, which is simple to implement and yet is accurate. The method uses overlapping control volumes, an idea mentioned in Hirsch[100], but scarcely investigated in literature. The convection-diffusion equation in Cartesian coordinates is solved, in domains of varying complexity.

2.2 Formulation

The conventional finite-volume method uses non-overlapping control volumes to discretize the computational domain. It uses linear or higher order polynomials to deter-

mine cell face values. On non-orthogonal grids, it is cumbersome to use higher order interpolations. The finite-element method with iso-parametric formulation can be used to overcome the above problem; however this method requires assembly of elemental matrices. In this work, we attempt to discretize the domain in such a way to retain the higher-order interpolation schemes similar to finite-element methods while avoiding the need for assembly like finite-volume methods.

Most of the extant finite-volume methods avoid overlapping control volumes by defining and using intermediate points. The values of the variables on these intermediate points are often obtained by line-interpolation from the neighbours. On non-orthogonal grids, it may be preferable to use higher-order finite-element type shape-functions for interpolation. To do this it is convenient to avoid intermediate points and, by accepting overlapping control volumes, to use the neighbouring points directly in the computations.

The solution domain is discretized into a structured non-orthogonal grid as shown in Figure 2.1. This can be done by any grid-generation package for boundary-fitted systems. A typical control volume is shown by the shaded area in the figure and also in Figure 2.2. This choice of control volume uses the grid point coordinates directly, without computing any intermediate points, to form control volumes. It can be seen that each interior grid-point has a control volume associated with it, of which it is the central node. Hence we can refer to these control volumes by the index of this central node, e.g., the control volume for (i,j) is shown in Figure 2.2. It can be seen that adjacent control volumes will overlap to some extent.

2.2.1   Governing Equations for a Control Volume

Now we apply the conservation laws to the typical control volume to get the algebraic discrete equations. The conservation form of the two–dimensional steady–state convection–diffusion equation for a scalar ϕ is

$$\nabla.(\rho \tilde{U} \phi) \ = \ \nabla.(\Gamma \nabla \phi) + S_\phi , \tag{2.1}$$
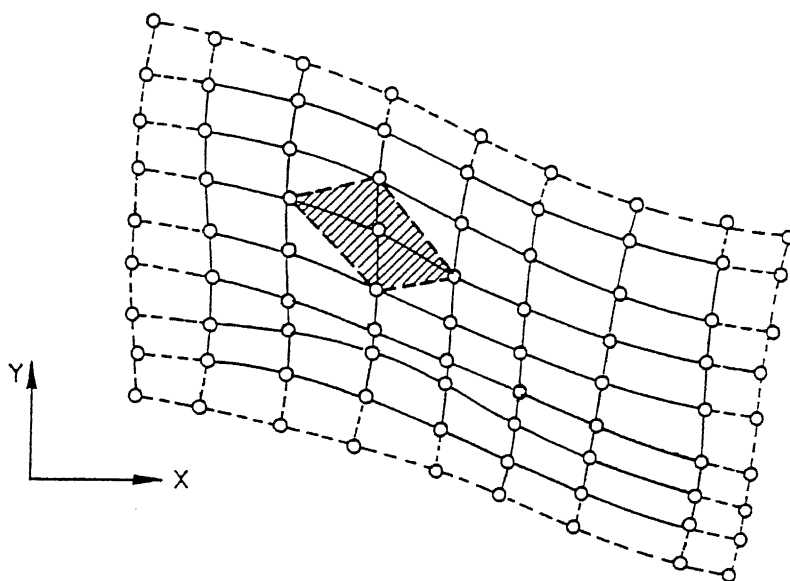
Figure 2.1: Schematic diagram
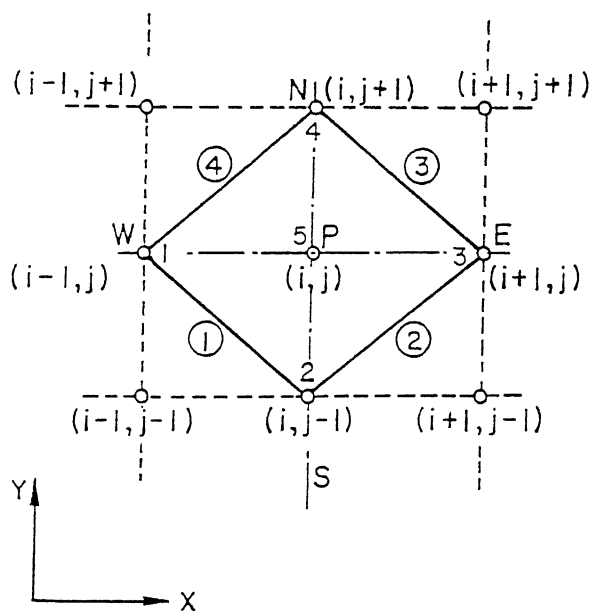
Figure 2.2: A typical control volume
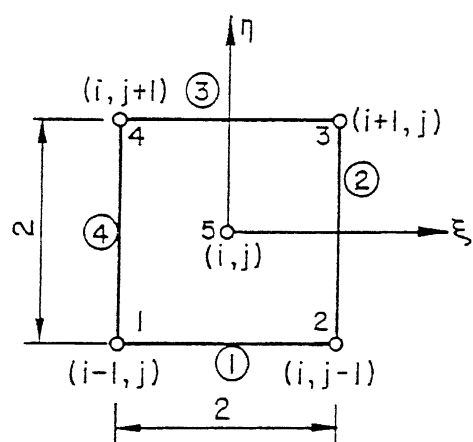
Figure 2.3: Iso-parametric Element

where ρ is the density, $\tilde{U}$ is the velocity vector having components u and v in the directions x and y, respectively, Γ is the diffusion coefficient, and S_ϕ is a source term. On integrating the equation (2.1) over the control volume and applying the Gauss–divergence theorem, we get

$$\oint_{cs} \phi(\rho u n_x + \rho v n_y)dl \;=\; \oint_{cs} \Gamma(\frac{\partial \phi}{\partial x}n_x + \frac{\partial \phi}{\partial y}n_y)dl + \int\int S_\phi dA \qquad (2.2)$$

where dl is an elemental length on the boundary (cs) of the control volume, n_x and n_y are the direction cosines of the outward normal $\hat{n}$ of dl. The area (As) of the control volume (i,j) in Fig 2.2 is calculated using the formula

$$A_s \;=\; 0.5[(x_{i,j+1} - x_{i+1,j})(y_{i,j-1} - y_{i+1,j}) - (x_{i,j-1} - x_{i+1,j})(y_{i,j+1} - y_{i+1,j})$$
$$+(x_{i,j-1} - x_{i-1,j})(y_{i,j+1} - y_{i-1,j}) - (x_{i,j+1} - x_{i-1,j})(y_{i,j-1} - y_{i-1,j})]$$

where $x_{i,j+1}, y_{i,j+1}$ etc are the coordinates of the neighbouring points shown in Figure 2.2.

The contour integration is counter–clockwise. The terms in the equation 2.2 are further approximated as follows :

2.2.2   Convection Term

Using the mid–point rule we approximate the convective term :

$$\oint_{cs} \phi(\rho u n_x + \rho v n_y)dl \;=\; \sum_{k=1}^{4} \phi^{(k)}(\rho u^{(k)}\Delta y^{(k)} - \rho v^{(k)}\Delta x^{(k)})$$
$$=\; \sum_{k=1}^{4} \phi^{(k)} F^{(k)} \qquad (2.3)$$

where the superscript (k) refers to the edges of the control volume (shown circled in Figure 2.2). For the edge (k), k=1,2,3 , the approximation used is (assuming constant density, ρ) :

$$u^{(k)} \;=\; 0.5(u_k + u_{k+1})$$
$$v^{(k)} \;=\; 0.5(v_k + v_{k+1})$$
$$\Delta y^{(k)} \;=\; (y_{k+1} - y_k)$$
$$\Delta x^{(k)} \;=\; (x_{k+1} - x_k)$$

where the subscript k refers to the local grid point number [shown in Figure 2.2]. For $k=4$, u_{k+1}, v_{k+1} etc. are replaced by u_1, v_1 etc., respectively, in the above equations.

The outward mass flux through the edge k is

$$F^{(k)} = (\rho u \Delta y - \rho v \Delta x)^{(k)} \tag{2.4}$$

To incorporate upwinding, $\phi^{(k)}$ in (2.3) is approximated at the mid-point of control-surface k by interpolation within the appropriate control volume depending on the flow direction across that surface. For example, if $F^{(1)}$ is negative (i.e. flow is entering into the control volume across face 1), then $\phi^{(1)}$ is approximated by interpolation within the control volume for node $(i-1, j-1)$. That is, the values at the grid points constituting the control volume $(i-1, j-1)$, through which the flow enters the control volume (i, j), are used for the interpolation of $\phi^{(1)}$ at the center of the surface 1. Otherwise, if $F^{(1)}$ is positive, the values at the grid points of control volume (i, j) are used to interpolate $\phi^{(1)}$ at surface 1. This scheme for convective modelling is obviously conservative as it always uses the upwind control volume. The method used for interpolation is based on finite–element type shape functions, and is explained below.

2.2.3   Interpolation

The control volume (i, j) is mapped on to a square in (ξ, η) space as shown in Figure 2.3, with the (i, j) node at $(0,0)$ and the other nodes at the vertices $(\pm 1, \pm 1)$ respectively. The following shape functions are used for the interpolations

$$
\begin{aligned}
N_1 &= 0.25(-\xi - \eta + \xi\eta) + 0.125(\xi^2 + \eta^2) \\
N_2 &= 0.25(\xi - \eta - \xi\eta) + 0.125(\xi^2 + \eta^2) \\
N_3 &= 0.25(\xi + \eta + \xi\eta) + 0.125(\xi^2 + \eta^2) \\
N_4 &= 0.25(-\xi + \eta - \xi\eta) + 0.125(\xi^2 + \eta^2) \\
N_5 &= 1 - 0.5(\xi^2 + \eta^2)
\end{aligned}
\tag{2.5}
$$

The iso-parametric formulation is used and the dependent variable ϕ, as well as the coordinates x and y, in the control volume are represented as

$$\phi = \sum_{i=1}^{5} N_i \phi_i \tag{2.6}$$

$$x = \sum_{i=1}^{5} N_i x_i \tag{2.7}$$

$$y = \sum_{i=1}^{5} N_i y_i \tag{2.8}$$

where the x_i's and y_i's are the x and y coordinates of the 5 grid points respectively. The equations (2.7) and (2.8) connect the control volume in Figure 2.2 onto the transformed control volume in (ξ, η) coordinates shown in Figure 2.3.

For the purposes of upwinding, $\phi^{(k)}$ is found by using (2.6) to interpolate the value at the mid–point of the face k in the transformed control volume. For example, for face 1 we compute $N_1, N_2, ..., N_5$ at $\xi=0$, $\eta = -1$, and then use (2.6).

The derivatives of the dependent variable (ϕ) are defined as

$$\frac{\partial \phi}{\partial x} = \sum_{i=1}^{5} \frac{\partial N_i}{\partial x} \phi_i \tag{2.9}$$

$$\frac{\partial \phi}{\partial y} = \sum_{i=1}^{5} \frac{\partial N_i}{\partial y} \phi_i \tag{2.10}$$

where the derivatives of the shape-functions and the determinant of the Jacobian, respectively, are given by

$$\left\{ \begin{array}{c} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{array} \right\} = \frac{1}{|J|} \left[\begin{array}{cc} \frac{\partial y}{\partial \eta} & -\frac{\partial y}{\partial \xi} \\ -\frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \xi} \end{array} \right] \left\{ \begin{array}{c} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{array} \right\} ; \tag{2.11}$$

and

$$|J| \equiv \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi}, \tag{2.12}$$

where

$$\frac{\partial x}{\partial \xi} = \sum_{i=1}^{5} \frac{\partial N_i}{\partial \xi} x_i \tag{2.13}$$

$$\frac{\partial x}{\partial \eta} = \sum_{i=1}^{5} \frac{\partial N_i}{\partial \eta} x_i \tag{2.14}$$

$$\frac{\partial y}{\partial \xi} = \sum_{i=1}^{5} \frac{\partial N_i}{\partial \xi} y_i \tag{2.15}$$

$$\frac{\partial y}{\partial \eta} = \sum_{i=1}^{5} \frac{\partial N_i}{\partial \eta} y_i. \tag{2.16}$$

The derivatives of the shape functions with respect to ξ and η can be computed using equations (2.5). Once the derivatives are computed equations (2.13-2.16) are used to compute right hand side of equation (2.11) and hence the desired derivatives of the dependent variables using equations (2.9) and (2.10). These values are computed only once during the initialization process and stored for further calculations.

2.2.4 Diffusion Term

The control volume is in each case the control volume for (i, j). This term is also approximated using the mid-point rule. The discretized equation is represented as

$$\oint_{cs} \Gamma(\frac{\partial \phi}{\partial x} n_x + \frac{\partial \phi}{\partial y} n_y) dl = \sum_{k=1}^{4} \sum_{i=1}^{5} \Gamma(\frac{\partial N_i}{\partial x} |_{mid} \Delta y^{(k)} - \frac{\partial N_i}{\partial y} |_{mid} \Delta x^{(k)}) \phi_i \qquad (2.17)$$

where $\Delta x^{(k)}$ and $\Delta y^{(k)}$ are the same as defined for the convective discretization. The derivatives of the shape functions are evaluated at the mid-points, in (ξ, η) space, of the control surfaces. The summation is carried out over all the edges of the control volume. The procedure to evaluate the derivative terms has been explained above. It can be noted that this discretization for the diffusion term, is not conservative. However, we invariably found the final converged solution to satisfy closely the conservative property of the scalar. It can be shown that on a regular grid, the diffusion modelling in OCV is second–order accurate and has exactly the same error as the conventional five-point center-difference discretization.

2.2.5 Source Terms and Boundary Conditions

The source term S can be represented in the general form

$$S_\phi = S_c + S_p \phi \qquad (2.18)$$

where S_c and S_p are stored at cell center and are assumed to prevail over the entire control volume.

In the proposed scheme, fictitious boundaries are created, along the domain boundary, as shown in Figure 2.1 by dashed lines. The values of the scalar ϕ at these fictitious

points are specified using quadratic extrapolation. These additional grid points are needed for the upwinding and diffusion term calculations at the grid points on the boundary (for Neumann boundary conditions) or next to the boundary (for Dirichlet boundary conditions).

2.2.6   Solution Procedure

Finally, with the above formulation, the discretized equation for the convection-diffusion equation (2.1) can be written as

$$a_p\phi_p = \sum_{nb} a_{nb}\ \phi_{nb} + b \qquad (2.19)$$

where ϕ_p is the (unknown) scalar value at the central node and ϕ_{nb} the (unknown) values at the neighbours (including those for neighbouring control volumes introduced by upwinding). The coefficients a_p and a_{nb} are given in the Appendix-A.

The coefficient matrix may lose its diagonal dominance in highly convective flows and the iterative scheme may thus become unstable. To facilitate iterative convergence, the terms with negative coefficients, are removed from the summation in (2.19) and are included in b. The Gauss-Seidel iteration technique is used to solve the discretized equation. Except for very mild overshoots and undershoots, for problems with step input and at high Peclet number, no major difficulties were encountered in solving the variety of test problems reported in this chapter.

2.3   Results

A number of steady-state convection-diffusion problems are now solved by the Overlapping Control Volume (OCV) method and the results are compared with those of other schemes. These test problems and the results obtained are discussed in this section. In all cases the governing equation is (2.1), with $\rho = 1$. The computed solutions are compared with the exact solution with the RMS error defined as below

$$E_{rms} = 100(\frac{\sum^M \mid \phi_{exact} - \phi_{computed}\mid^2}{M})^{1/2}$$

where the summation is over all M interior points of the domain.

When other schemes are used, for comparison, in all cases the method of iteration is the same: Gauss-Seidel using a *positive-coefficient only* matrix, with the coefficients in equation (2.19) varying with scheme used. All computations were done on a DEC-3000 computer.

2.3.1   Test Problem 1

This test problem is used to show the efficacy of the convection modelling of the OCV method. The comparisons with the power-law scheme and QUICK[6] are presented, for a range of grid levels. The computational domain and boundary conditions are shown in Figure 2.4. The velocity components are defined as $u = -y$ and $v = x$. If the problem is purely convective ($\Gamma = 0$), then any scalar profile specified along OA in Figure 2.4 should be swept unchanged along the streamlines and reproduced at OB (after going through a 270° turn).

The computational domain is discretized using uniform grids of dimension $N \times N$ with N= 31, 41, 51, 61, 71, 81. The diffusion coefficient $\Gamma = 10^{-6}$, $\rho = 1$, and L=2 are used in the computations. A smooth profile specified along OA is the Gaussian distribution : $\phi = e^{2|x|} sin^2(\pi x)$. The percentage RMS error for the points on OB (*not* for the entire domain) for power-law, QUICK and OCV at different grid levels are shown in Figure 2.5 on log-log scale. It can be seen that the percentage RMS error for the OCV scheme is much less than that for the power-law scheme. However, the performance of QUICK is best in this case. To estimate the order of accuracy, for a smoothly varying solution, the error level ϵ can be assumed to vary in proportion to Δx^m where m is the order of accuracy. The exponent for the power-law, QUICK and OCV schemes are found to be 0.606, 2.004 and 2.054, respectively. This shows that the convection modelling used in the OCV scheme is, like QUICK, second order accurate. The error of OCV is slightly larger than that of QUICK.

The *CPU* time taken by various schemes obviously play some part in our estimation of them. We compare E_{rms} *Vs CPU* time in Figure 2.6 for the same cases. We find
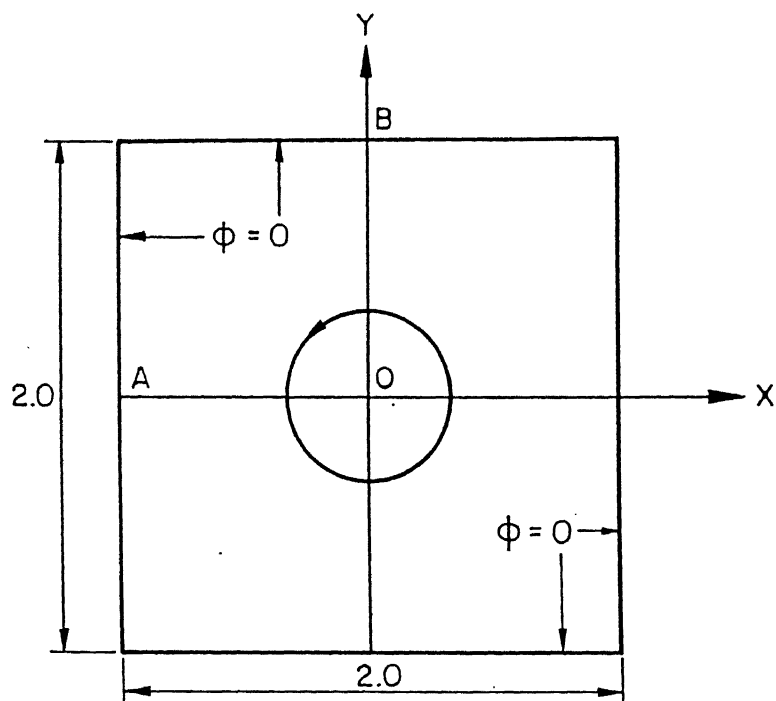
Figure 2.4: Schematic diagram of test problem 1
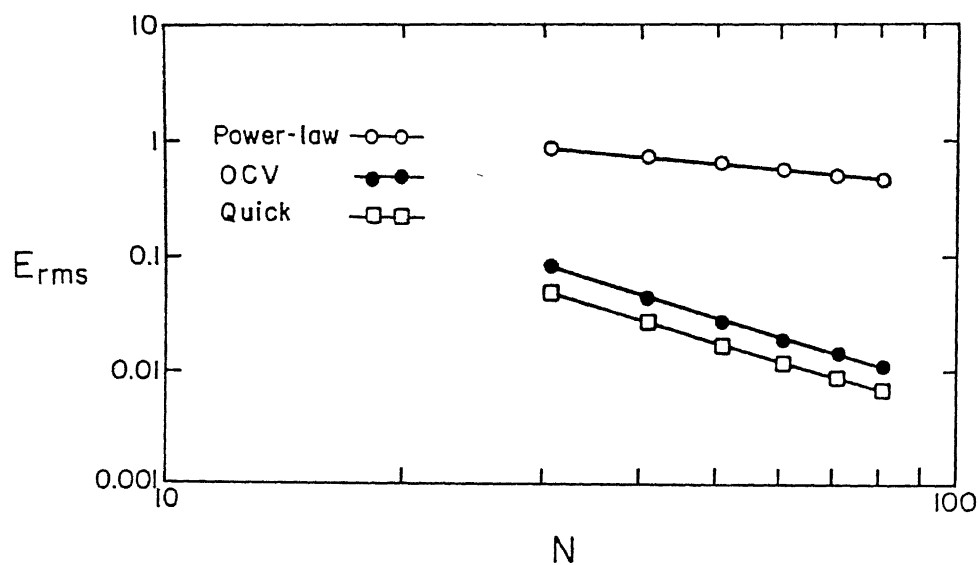
Figure 2.5: RMS error versus N for test problem 1
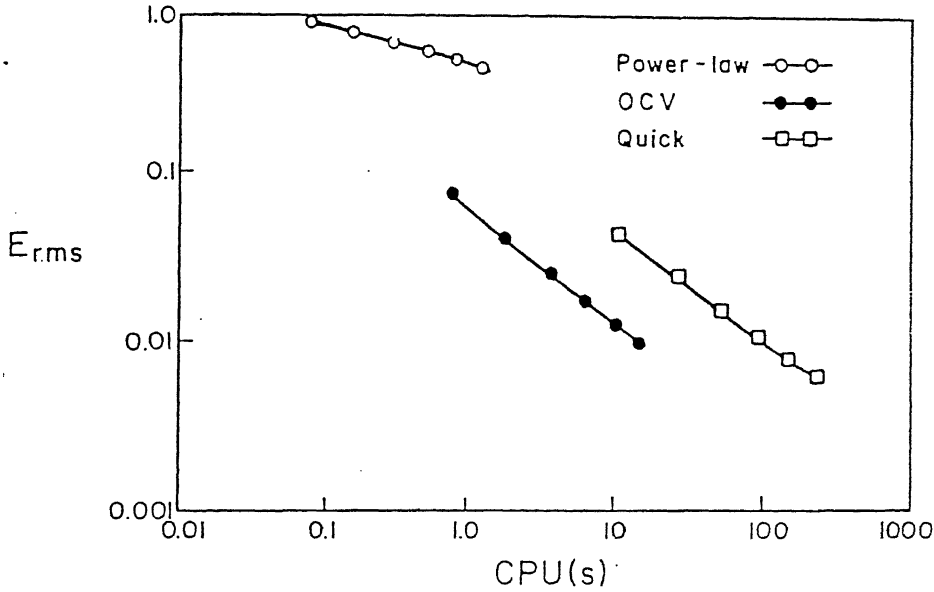
Figure 2.6: RMS error versus CPU-time for test problem 1

OCV, because it converges faster than QUICK, gives better accuracy for a given CPU time. The power-law performs relatively poor in comparison to the other two schemes.

2.3.2   Test Problem 2

This problem, used by Runchal[101] , is shown in Figure 2.7. The inner and outer surfaces of a cylindrical annulus are maintained at constant ϕ_1 and ϕ_2 (assumed as 1 and 0 respectively). The Cartesian velocity components are represented as $u = 2r^{n-1}y$ and $v = -2r^{n-1}x$. Two values of the constant n are considered: $n = 1$, representing solid-body rotation with angular velocity $\omega=2$, and $n = 3$, for which the angular velocity is $\omega=2r^2$.

The problem is one–dimensional in cylindrical coordinates, hence its exact solution can be obtained easily. However, in Cartesian coordinate system, this problem becomes two-dimensional. The exact solution, for all Peclet numbers and values of n, is given
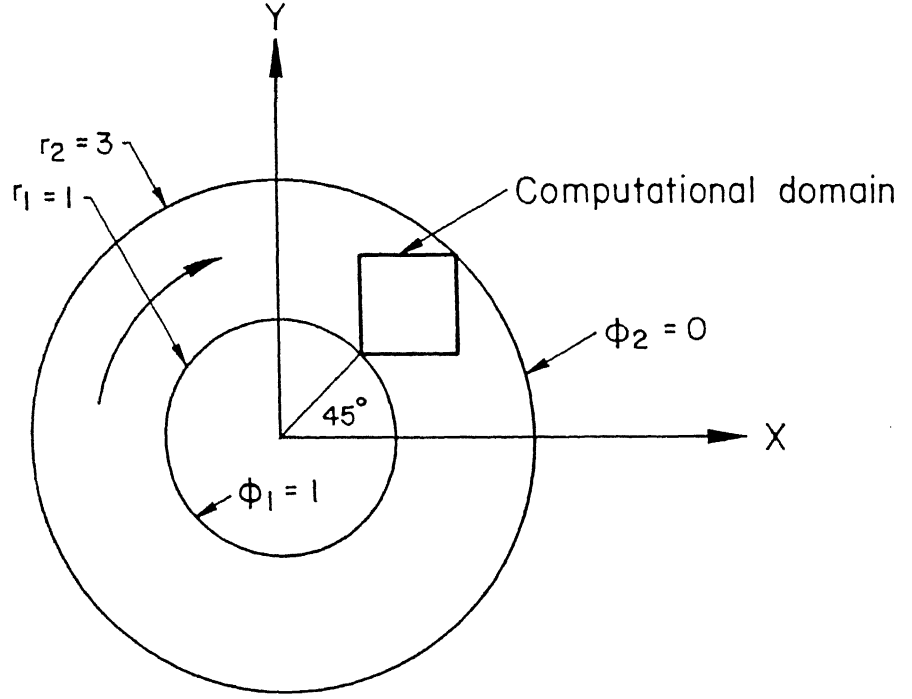
Figure 2.7: Schematic diagram of test problem 2

by

$$\phi = 1 - \frac{\ln(r/r_1)}{\ln(r_2/r_1)},$$

where r_1 and r_2 are, respectively, the inner and outer radii of the annulus.

The computational domain is the square shown in the figure, and is divided into $N \times N$ grid points. In this problem, we take $r_1 = 1$, and $r_2 = 3$. The exact solution is used to specify the Dirichlet boundary conditions along the boundaries of the square domain.

We compare the results of OCV scheme with the power-law and QUICK schemes for this problem, for a range of Peclet numbers from 0.01 to 10,000, where $Pe \equiv \omega r_1^2/\Gamma$. It should be noted however this is only the nominal Peclet number (based on the inner

radii of the annulus). The true Peclet number $V\sqrt{2}/\Gamma$ (the side of the square $=\sqrt{2}$) is a variable and its value at the mid-point of the domain is $2\sqrt{2}$ and $8\sqrt{2}$ times the nominal value for n=1, and 3, respectively.

The Figure 2.8 shows the percentage RMS error for $n = 1, 3$. The OCV scheme shows better results than the power-law scheme but QUICK is best for this case. For low Peclet numbers, the schemes shows nearly the same error, as the diffusion modelling in power-law and QUICK is the same center-difference scheme, and that of OCV is equivalent.

In Figure 2.8, for $n = 3$, the percentage RMS error for the power-law scheme increases very rapidly with Peclet number whereas E_{rms} for QUICK and OCV remains almost same as that for $n = 1$. Figure 2.9 can be used to estimate the order of accuracy of the schemes for a range of Peclet numbers, Pe= 10, 100 and 1000. The value of n, used to specify velocity fields is 1. Five different grid levels, N=11, 21, 31, 41 and 51 are used. It can be seen that the slope of the E_{rms} error line for the OCV is nearly same as that for QUICK, and the accuracy is much better than that of the power-law scheme. The exponent m, same as defined in Test Problem 1, for the power-law, OCV and QUICK are found to be 1.65, 2.25, 2.40 for Pe= 10; 0.88, 2.28, 2.55 for Pe= 100; and 0.88, 2.27, 2.56 for Pe= 1000; respectively. Once again OCV demonstrates that it is a second order method. QUICK is somewhat better than OCV at all Peclet numbers, for the same grid size.

However, we compare E_{rms} Vs CPU in Fig 2.10 for Pe=100, n=3, which is typical, and once again find OCV slightly better than QUICK for the same CPU time.

2.3.3   Test Problem 3

This problem is designed to test the applicability of the proposed scheme to arbitrary geometries. The details and exact solution of this problem are the same as those of Test problem 2. The only difference lies is the interior grid points are randomly perturbed from their original uniform grid positions to a maximum extent of 10% of the average grid distance (i.e. each interior grid point is shifted from the uniform grid position by
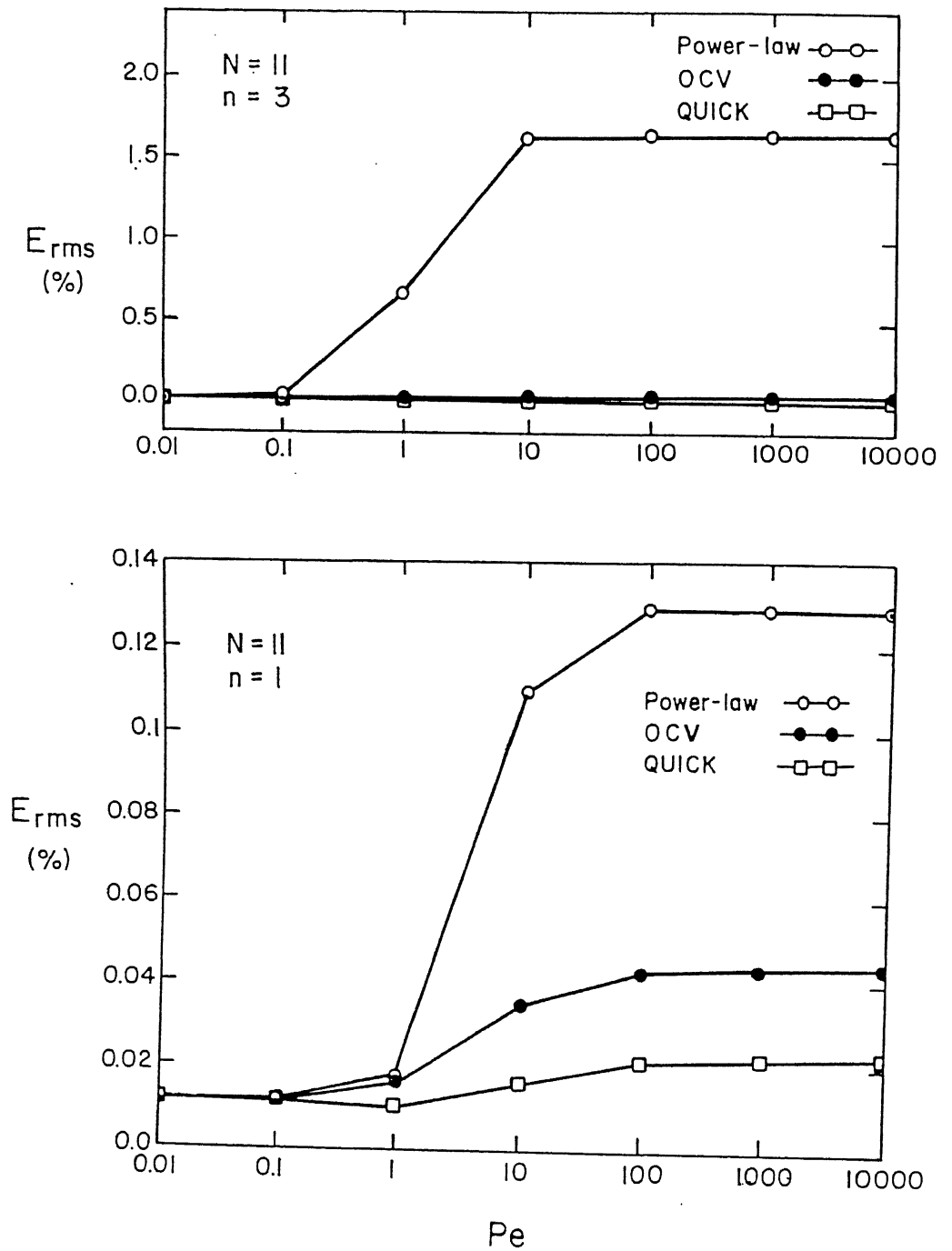
Figure 2.8: RMS error versus Pe for test problem 2
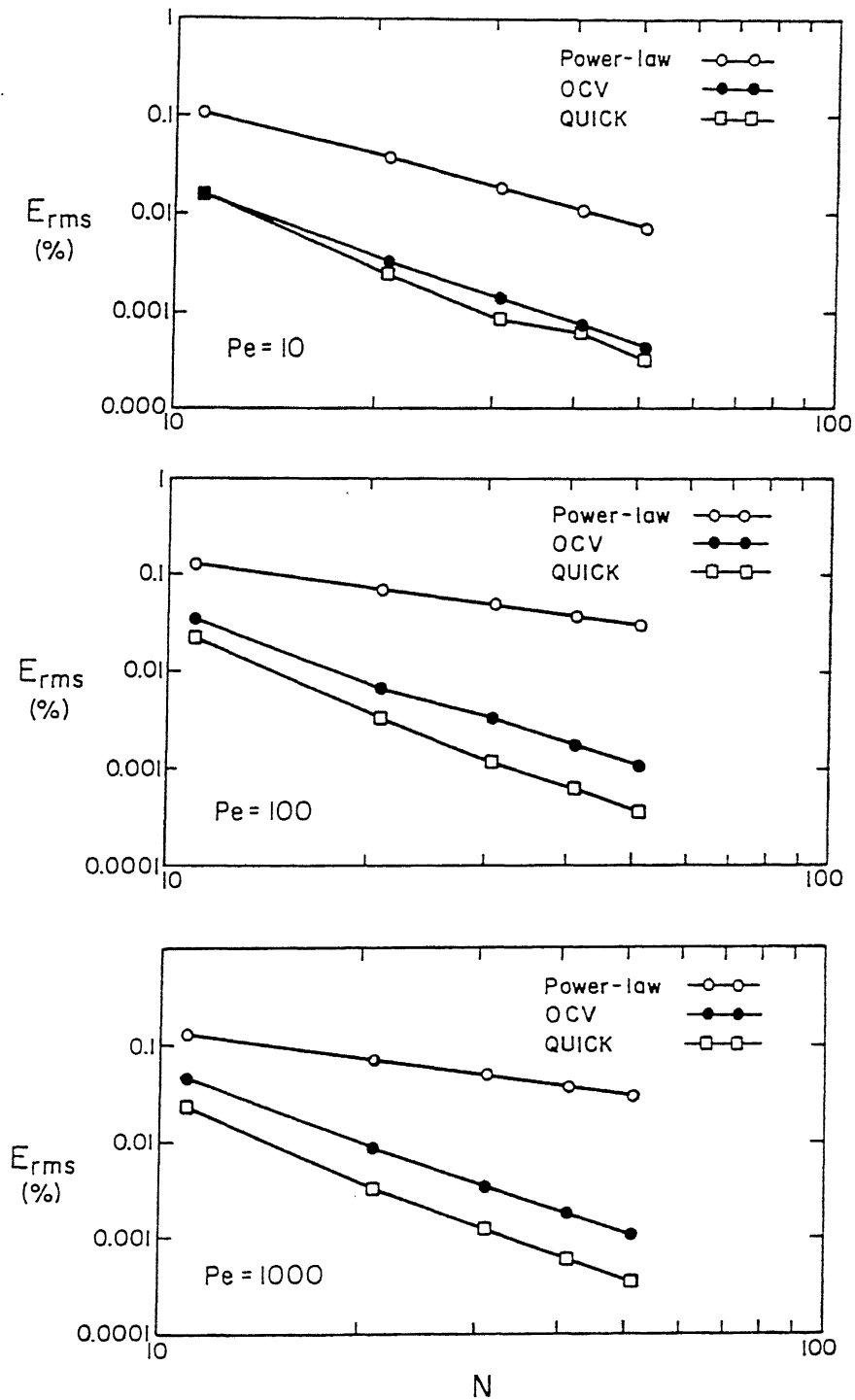
Figure 2.9: RMS error versus N for test problem 2 ($n=1$)
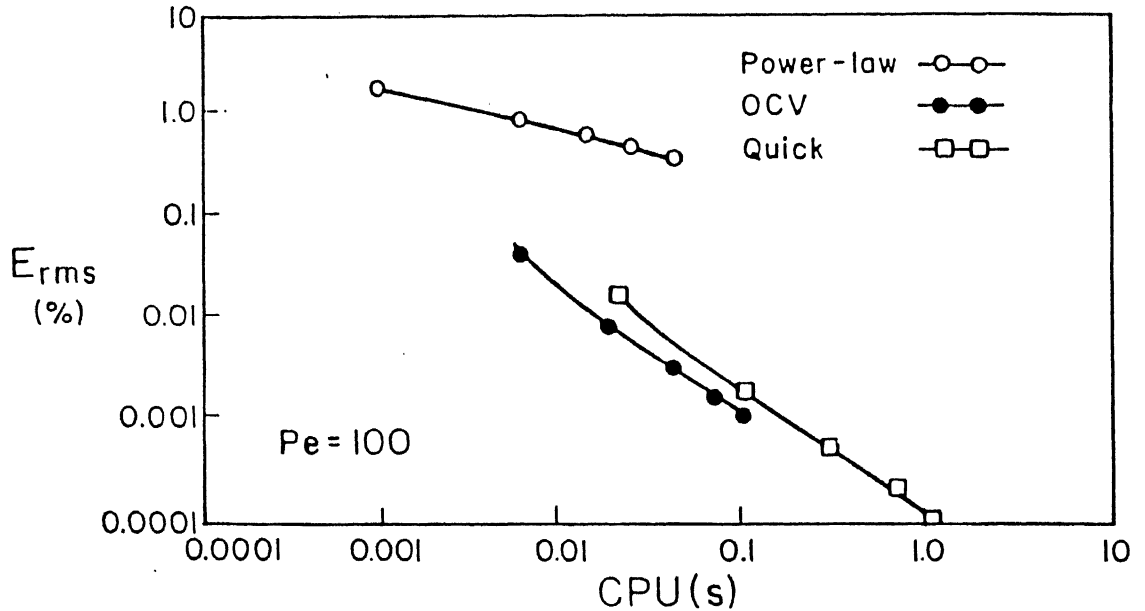
Figure 2.10: RMS error versus CPU-time for test problem 2 (P_e=100, n=3)

Δx_s, Δy_s in the x and y directions, where Δx_s and Δy_s are uniformly distributed random numbers lying between $\pm(\%distortion) \times \Delta x/100$ and $\pm(\%distortion) \times \Delta y/100$, respectively). The domain and grid for $N = 11$ is shown in Figure .

We now consider four schemes for this problem: The power-law and QUICK schemes (both with the conventional center-difference diffusion modelling, see Ref. 12) are used for the transformed equation generated by mapping the grid shown in Figure 2.11 onto a rectangular grid. This is simply to map the points $x(i,j)$, $y(i,j)$ onto $i\Delta\xi$ and $j\Delta\eta$ on the (ξ,η) plane. The governing equation is also transformed (see Ref. 12) and solved on the (ξ,η) plane. The other two methods we consider are those of Hwang[102] and OCV which directly apply on the grid in the physical plane. The diffusion modelling presented by Hwang[102] is used here along with the power-law method for convection.

Figure 2.12 shows a comparison of the errors obtained on N= 11 grid for n= 1 and 3. It can be seen that at Peclet numbers atleast upto 100, the OCV scheme gives the best results. This result is repeated for values of N= 21, 31, 41, and 51 and also for grid distortions of 3% and 5%.
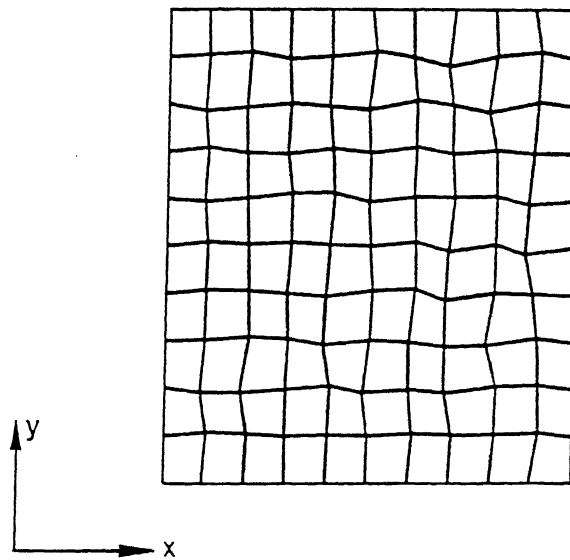
Figure 2.11: Distorted (10 per cent) grid for test problem 3
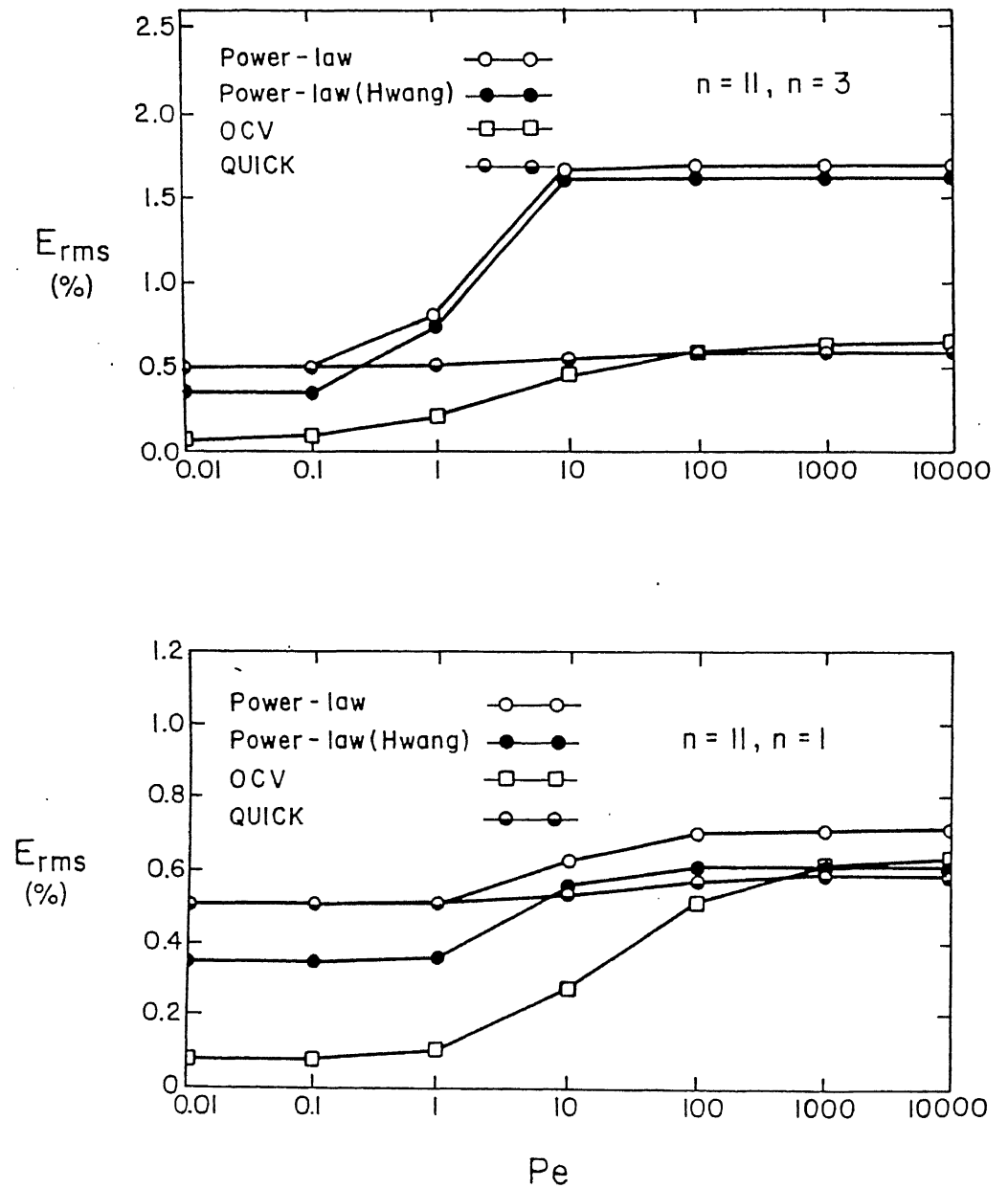
Figure 2.12: RMS error versus P_e for test problem 3 ($N=11$)
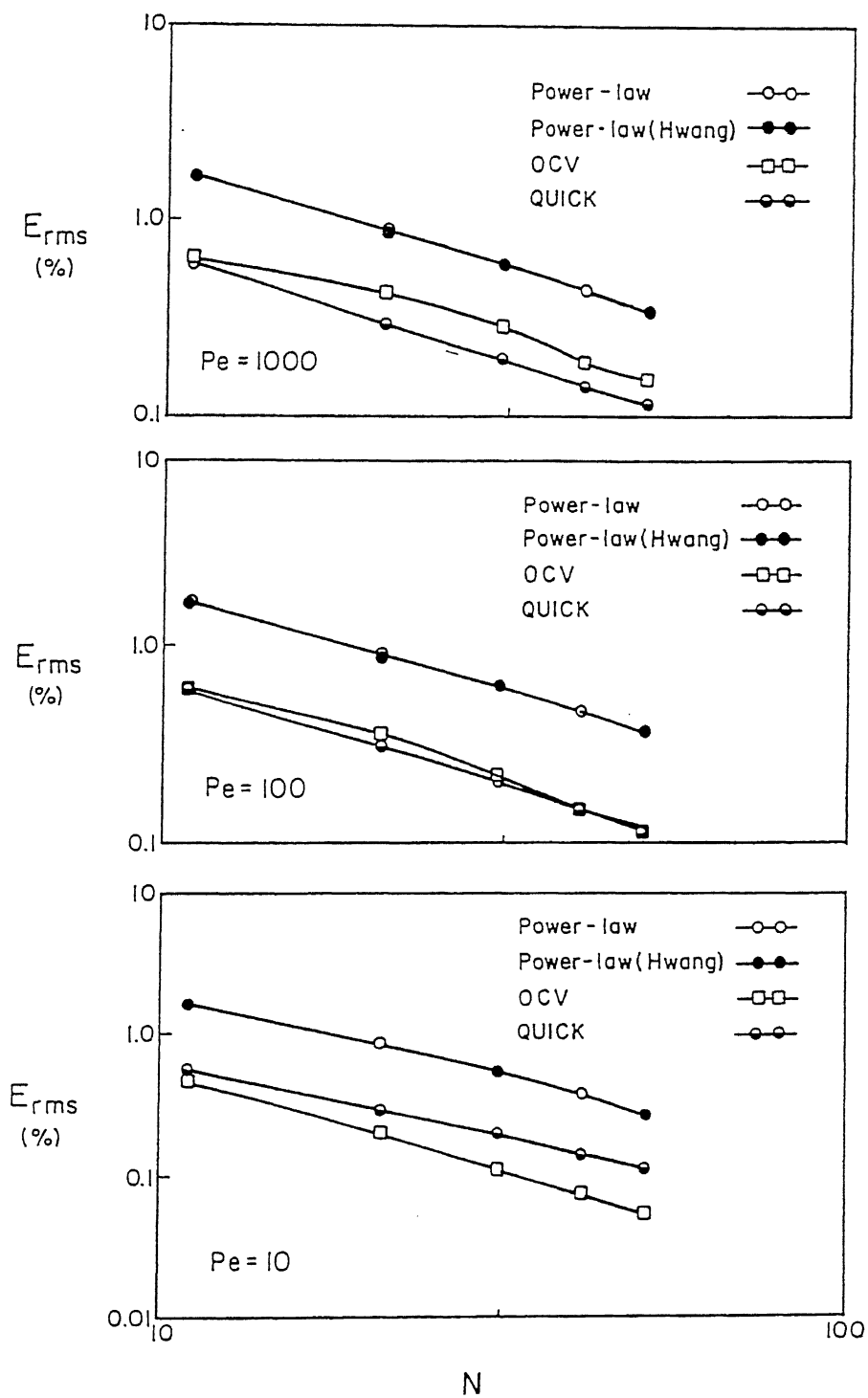
Figure 2.13: RMS error versus N for test problem 3 ($n=3$)
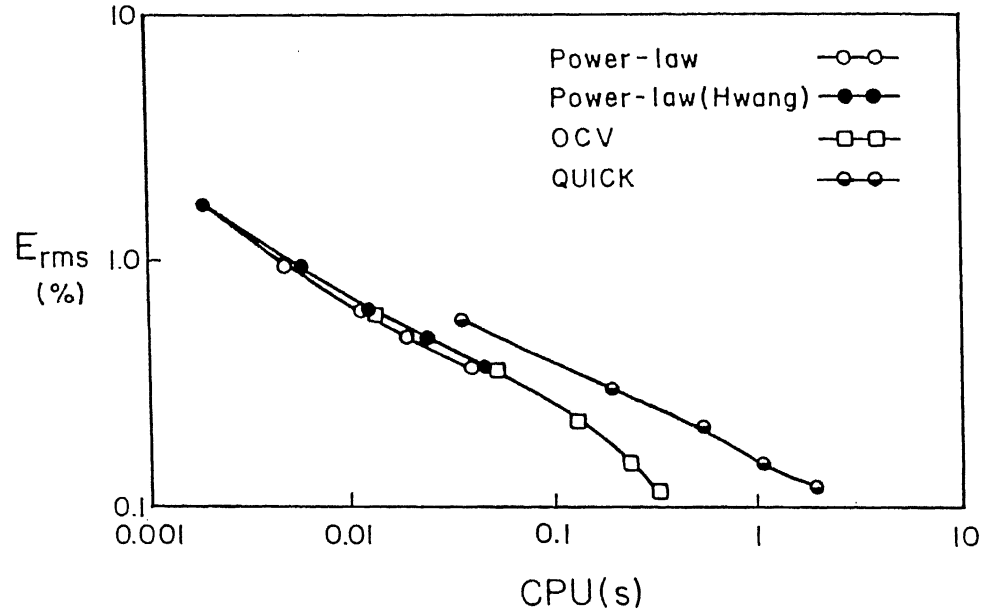
Figure 2.14: RMS error versus CPU-time for test problem 3 (P_e=100, n=3)

Figure 2.12 also shows (at the limit of Pe $\rightarrow$ 0) that the OCV diffusion modelling is superiour to Hwang[102] which in turn is superiour to the (transformed) center-difference diffusion modelling. It is clear that the Laplacian term introduces error in the schemes for the transformed equations which has a deletrious effect on their accuracy except at high Peclet numbers (remember that the actual Peclet number of the problem is higher than the nominal values used in the figures). This conjecture was confirmed by studying the case of pure diffusion on the distorted grid. It was seen that the error in the schemes relying on transformation increases very rapidly with the degree of distortion in the grid. Figure 2.13 shows the E_{rms} vs N for the four schemes for Pe= 10, 100, and 1000 for n=3. In all these cases the two power-law schemes give approximately same results. It can be seen that for Peclet numbers of 10, the accuracy, and, the improvement with N, is best for OCV. For Pe= 100, OCV and QUICK are comparable. Only above that Peclet number does the QUICK (on transformed domain) perform better. Therefore on distorted grids it may be supposed that OCV has a better performance in the range of practical grid Peclet numbers.

The E_{rms} Vs CPU time is shown in Figure 2.14 for Pe= 100, n=3, a typical case. The

Table 2.1: CPU Time and (Number of Iteration for Convergence) for Pe= 10, n=3

GRID	Power-law(Hwang[102])	OCV	QUICK	Power-law
11× 11	0.00195 (5)	0.00585 (24)	0.03025 (34)	0.00195 (5)
21× 21	0.00976 (11)	0.02342 (22)	0.18446 (41)	0.00976 (11)
31× 31	0.03025 (21)	0.04684 (18)	0.50849 (50)	0.03025 (21)
41× 41	0.07515 (35)	0.09174 (19)	1.03356 (66)	0.08003 (35)
51× 51	0.16299 (47)	0.24490 (47)	2.43122 (87)	0.18153 (46)

OCV scheme does better than QUICK. The power-law schemes, both on transformed and physical domains, also do well, presumably because their superior convergence properties outweigh their first order accuracy.

The *CPU*-times (in seconds) are shown in Table 2.1 for the various schemes, at different N. The quantities in brackets are the iterations needed to reach convergence (residual of 10^{-7}, with double-precision arithmetic). QUICK is seen to be expensive and becoming increasingly so with grid refinement, taking not only more *CPU* time per iteration but also requiring more number of iterations. It must be mentioned that the relatively poor performance of QUICK in the *CPU* time computations, can be attributed directly to the poor convergence properties of the coefficient matrix of the scheme used in equation (2.19). This matrix will not be used in explicit time-stepping, and will be different for implicit time stepping. Therefore, the performance of QUICK here is uniquely for steady-state solutions, and extrapolating to unsteady solutions would be injudicious.

2.3.4   Test Problem 4

This model problem, first proposed by Raithby[15] , is widely used to test the cross-stream numerical diffusion of difference schemes.

The problem is shown in Figure 2.15. The flow is assumed uniform and passing through the square domain making an angle such that the streamline through (0, y_c) passes through the center of the square. The scalar field at the left (inflow) boundary has an abrupt step change, with ϕ=1 above y_c and ϕ=0 below. The Peclet number is taken as
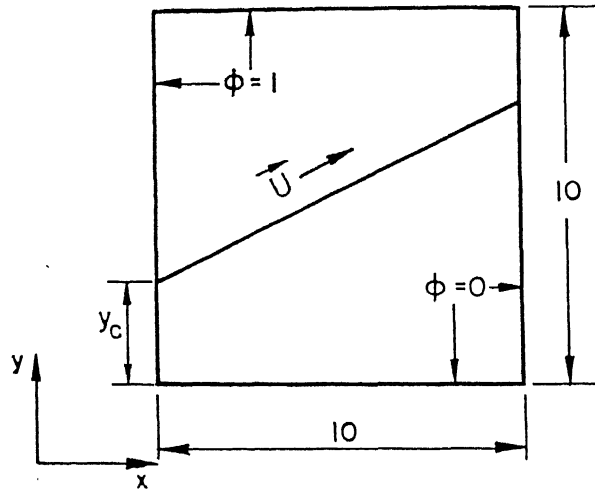
Figure 2.15: Schematic diagram of test problem 4

infinity and thus the scalar property is transported by convection only. The boundary conditions are shown on the figure. The value of ϕ is assumed 0.5 at the point of the step change, $y=y_c$, $x=0$. The exact solution for this problem is the convection of the step-change in the flow direction without any diffusion. Thus $\phi=1$ above the slanted line shown and $\phi=0$ below it.

An regular 11 $\times$ 11 grid is used to discretize the solution domain. The results at the mid plane ($x=5.0$) are shown in Figures 2.16, 2.17, 2.18 for $\Gamma = 10^{-6}$ and $y_c=0$, 3, 5, respectively. The results of the OCV scheme are compared with those of power-law and QUICK schemes. Figure 2.16 shows that the OCV gives minimum smearing errors compared to all other schemes, at $y_c = 0$. This is the worst case for all other schemes. For $y_c = 3$, it can be seen from Figure 2.17 that the level of overshoots and undershoots are approximately of the same order for the OCV and QUICK. The power-law scheme again gives maximum cross-wind diffusion but there is no overshoot or undershoot. At $y_c = 5$, when flow is aligned to grid lines, it can be observed from Figure 2.18 that all other schemes except OCV gives exact solution. The OCV scheme introduces small overshoots and undershoots in this case.
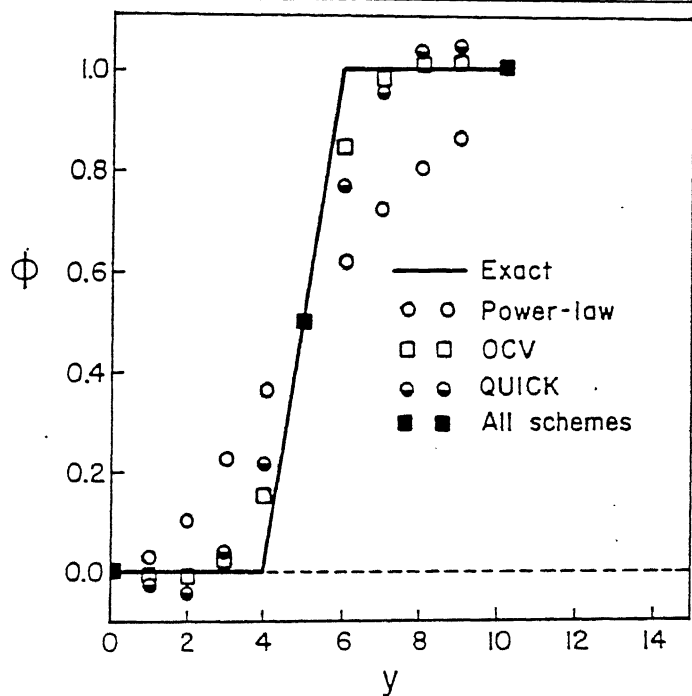
Figure 2.16: Results of test problem 4 for $y_c=0$
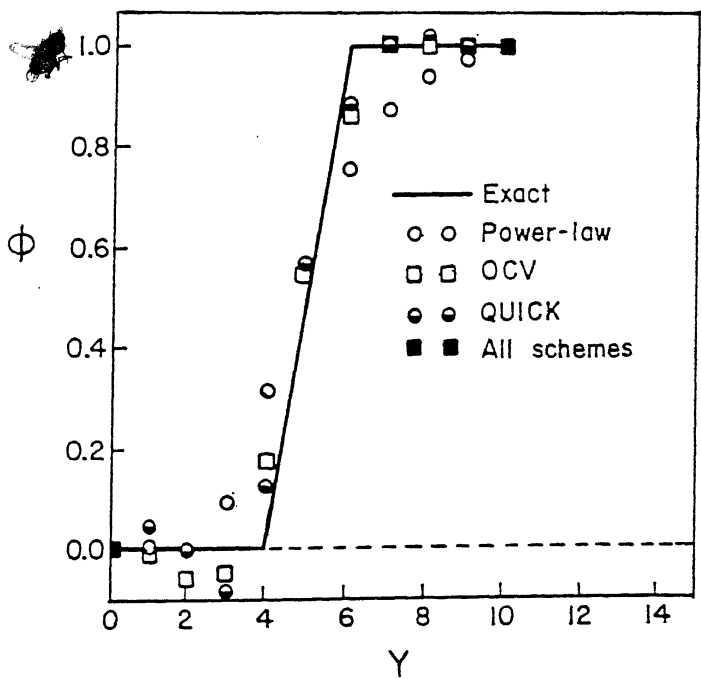
Figure 2.17: Results of test problem 4 for $y_c=3$
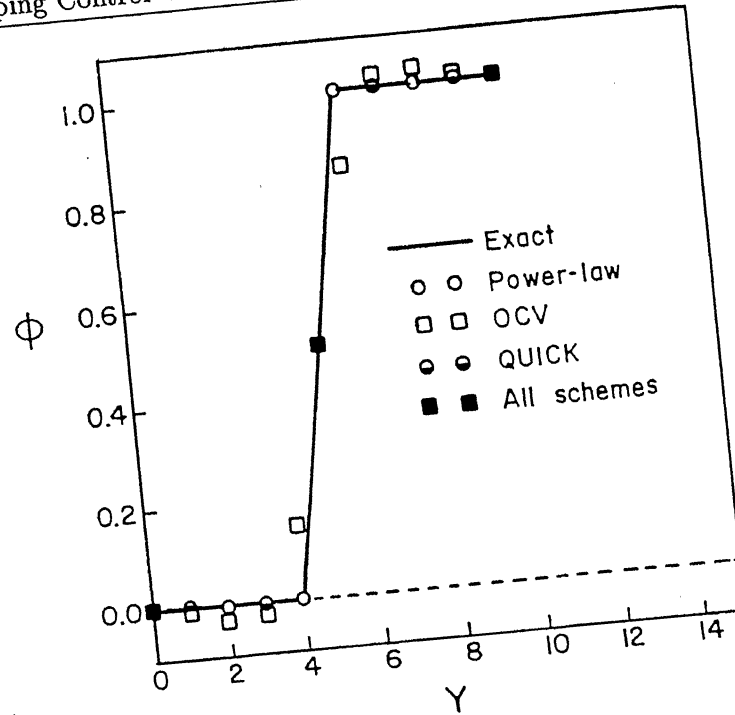
Figure 2.18: Results of test problem 4 for $y_c=5$

Thus OCV introduces cross-stream numerical-diffusion, except for the case when flow is aligned to the grid line, at levels below that of power-law scheme and comparable to QUICK. It however, like QUICK and other higher-order schemes, allows overshoots and undershoots.

2.4   Summary

The following observations can be made on the basis of results obtained above. The proposed OCV scheme performs consistently well in all the test cases.

- This chapter introduces a finite-volume method to solve steady–state convection–diffusion 2-D problems on structured non-orthogonal grids.

- Overlapping control volumes (OCV) are used to discretize the physical domain, and the governing equations are solved without transformation. An iso-parametric formulation is used to compute diffusion and for upwinding.

- The OCV treatment of diffusion seems to be quite effective, even on distorted meshes. On uniform meshes, it is second-order accurate.

- For convection dominated flows, the OCV treatment of the convection term, like that of the QUICK scheme, is second order accurate on an uniform mesh. Although the OCV scheme does not perform well as QUICK, for the same grid size, it does better for the same *CPU*-time.

- When both, convection and diffusion, processes are significant, the results obtained using OCV scheme is quite encouraging on non-orthogonal meshes. It does much better than the power-law scheme. Its accuracy for the non-orthogonal test problem attempted is comparable to the QUICK scheme, used with center-difference modelling for the diffusion term, while it is computationally less expensive.

- On non-orthogonal grids, OCV gives better accuracy for a large and practical range of Peclet numbers than does QUICK applied to the transformed equations using the conventional five-point diffusion modeling.

- The results obtained also demonstrate that the scheme reduces false-diffusion to a considerable extent in comparison with the power-law scheme.

Chapter 3

A Flux Limiter for the OCV Method

3.1 Introduction

In this chapter, we present a flux limiting scheme FLOCV for the overlapping control volume (OCV) approach for 2-D steady and unsteady convection – diffusion problems on structured non-orthogonal grids. FLOCV switches from second to first-order interpolation in the presence of extrema. Smooth switching between the two is ensured by weighted averaging of second-order and first-order upwind differencing, with the weights being dynamically determined. Five Test problems are solved using this scheme and the results are compared with known analytical solutions. It is found that the FLOCV approximately retains the second-order of accuracy of the base discretization scheme on uniform grids and smooth non-uniform orthogonal grids. It is also found effective in removing oscillations for problems with discontinuities on both orthogonal and non-orthogonal grids, with little degradation of accuracy.

3.2 Formulation

The details of the OCV formulation for the steady-state convection-diffusion equation are given in chapter 2. We now solve the unsteady convection-diffusion equation by

an explicit time-stepping scheme. The discretization for the convection and diffusion terms are same as that presented in chapter 2. We present below the discretization of the transient term only, and change in the solution procedure, to avoid unnecessary *repetition.*

3.2.1   Governing Equations for a Control Volume

The conservation form of the two–dimensional time-dependent convection–diffusion equation for a scalar ϕ is

$$\frac{\partial(\rho\phi)}{\partial t} + \nabla.(\rho\tilde{U}\phi) = \nabla.(\Gamma\nabla\phi) + S_\phi \tag{3.1}$$

where ρ is the density, $\tilde{U}$ is the velocity vector having components u and v in the directions x and y, respectively, Γ is the diffusion coefficient, and S_ϕ is a source term. On integrating the equation (3.1) over the control volume and applying the Gauss divergence theorem, we get

$$\frac{\partial}{\partial t}\int_{cv}\rho\phi dA + \oint_{cs}\phi(\rho u n_x + \rho v n_y)dl = \oint_{cs}\Gamma(\frac{\partial\phi}{\partial x}n_x + \frac{\partial\phi}{\partial y}n_y)dl$$
$$+ \int\int S_\phi dA \tag{3.2}$$

where dl is an elemental length on the boundary (cs) of the control volume, n_x and n_y are the direction cosines of the outward normal $\hat{n}$ of dl. The contour integration is counter–clockwise.

The transient term is discretized as

$$\int_{cv}\rho\frac{\partial\phi}{\partial t}dA \simeq \rho(\frac{\phi_{i,j}^{n+1} - \phi_{i,j}^n}{\Delta t})A_s \tag{3.3}$$

where A_s is the area of the control volume and n and $n+1$ are time-step indices. The other terms can be evaluated at $n+1$ (for a fully implicit scheme) or at n (for an explicit scheme) or as a linear combination of the two. In this chapter, we use only the explicit scheme for the unsteady test cases.

The discretization of the convection and diffusion terms is given in detail in chapter 2. The discretizations use five-point shape functions and incorporate second-order

upwinding in the convective term. As the convective term discretization is critical to the exposition of the flux limiter we present it briefly below :

Using the mid–point rule we approximate the convective term :

$$\oint_{cs} \phi(\rho u n_x + \rho v n_y)dl = \sum_{k=1}^{4} \phi^{(k)}(\rho u^{(k)}\Delta y^{(k)} - \rho v^{(k)}\Delta x^{(k)})$$

$$= \sum_{k=1}^{4} \phi^{(k)} F^{(k)} \qquad (3.4)$$

where the superscript (k) refers to the edges of the control volume (shown circled in Figure 2.2). For the edge (k), $k=1,2,3$, the approximation used is (assuming constant density, ρ) :

$$u^{(k)} = 0.5(u_k + u_{k+1})$$
$$v^{(k)} = 0.5(v_k + v_{k+1})$$
$$\Delta y^{(k)} = (y_{k+1} - y_k)$$
$$\Delta x^{(k)} = (x_{k+1} - x_k)$$

where the subscript k refers to the local grid point number [shown in Figure 2.2]. For $k=4$, u_{k+1}, v_{k+1} etc. are replaced by u_1, v_1 etc., respectively, in the above equations.

The *outward* mass flux through the edge k is

$$F^{(k)} = (\rho u \Delta y - \rho v \Delta x)^{(k)}. \qquad (3.5)$$

To incorporate upwinding, $\phi^{(k)}$ in (3.4) is approximated at the mid-point of the edge k by interpolation within the *upwind* control volume, which is determined by the flow direction across the edge. That is, if $F^{(1)}$ is negative (i.e. flow is entering into the control volume across face 1), then $\phi^{(1)}$ is approximated by interpolation within the control volume $(i-1, j-1)$. Otherwise, if $F^{(1)}$ is positive, the values at the grid points of control volume (i,j) are used to interpolate $\phi^{(1)}$ at surface 1.

(ii) Second-order upwinding

The OCV method presented in chapter 2 uses second-order upwinding for the convective term. That is, the value of $\phi^{(k)}$ is obtained by using the relations (2.6), with the ϕ_i values of the upwind cell, and with ξ and η chosen appropriately. In chapter 2, this convective scheme was shown to be second-order accurate on a rectangular grid.

(ii) First-order upwinding

We can get a first-order estimate $\phi_1^{(k)}$ by simply equating the face value to the value at the central node of the upwind cell. For example, if $F^{(1)}$ is negative, then $\phi^{(1)}$ is approximated by $\phi_{i-1,j-1}$, or if $F^{(1)}$ is positive, by the value $\phi_{i,j}$.

Both these upwinding schemes for convective modelling are conservative. The method used for interpolation, in the second-order upwinding, is based on finite–element type shape functions, and was explained in chapter 2.

3.2.2 Boundary Conditions

Dirichlet boundary conditions can be implemented easily. For the control-volumes next to the boundary, second-order upwinding may require one extra (fictitious) grid-point outside the physical-domain. The solution procedure presented in this chapter avoids the need for this fictitious point by using first-order upwinding for these cell faces. For Neumann boundary conditions, backward or forward differencing can be used depending upon the boundary.

3.2.3 Solution Procedure

We solve the steady-state convection-diffusion equation by Gauss-Seidel iterations as described in chapter 2 and the unsteady convection-diffusion equation by the explicit method mentioned above.

3.3   Flux limiter

In common with most second-order schemes, OCV displays dispersion errors. Near
sharp gradients, the numerical solution has unphysical overshoots and undershoots. To
control these oscillations we propose the FLOCV (Flux Limited Overlapping Control
Volume) scheme, which uses a flux limiting procedure during every iteration of the
Gauss-Seidel routine or at each time step of the explicit scheme.

The algorithm that follows was partly inspired by Leonard's[34] Variable method. The
idea, roughly, is to use second-order upwinding as much as possible, to ensure good
accuracy, and lapse to first-order upwinding only for *abnormal* cells, where the use of
second-order upwinding might cause unboundedness. The degree of abnormality of a
cell is determined using a simple criterion, *i.e.*, by determining whether the scalar value
at the center node of a upwind cell is outside the range of values of the cell concerned.
It was found that too-violent a switching between first- and second-order upwinding led
to problems in convergence, so a smooth transition between the two was incorporated
using a parameter Δ (see below).

As mentioned above, the scalar face value $\phi^{(k)}$ in (3.4) can be estimated by either the
second-order upwind estimate, $\phi_2^{(k)}$, or the first-order one, $\phi_1^{(k)}$. These estimates, $\phi_2^{(k)}$
and $\phi_1^{(k)}$, are used in the flux limiting procedure. The magnitude of the overshoots
and undershoots in the solution can be controlled by blending of the second-order
scheme (OCV) and the first-order scheme. The blending is controlled by Δ, a switching
parameter. The implementation of FLOCV is straightforward for explicit schemes,
however for steady-state problems or implicit schemes with iterative procedures we
use a deferred-correction approach similar to Khosla and Rubin[103]. In this approach,
the quantity, $\phi^{(k)}$, computed by the procedure given below, is split in the following
manner :

$$[\phi^{(k)}]^{p+1} = [\phi^{(k)} - \phi_1^{(k)}]^p + [\phi_1^{(k)}]^{p+1} \tag{3.6}$$

where p and k are the iteration index and the face number, respectively. Here, the first
term on the right of equation (3.6) is known and is therefore included on the right-side
of equation (3.2), while the second term is unknown and has to be included in the
coefficients on the left side. This simple trick considerably improves the convergence

rate of the Gauss-Seidel iterations. For an explicit scheme $\phi^{(k)}$ is computed only once per time-step and is used directly.

The interpolated value $\phi^{(k)}$ is obtained by the following procedure designed to control oscillations :

We define, for each control volume (i,j)

$$\phi_{min} = min(\phi_{i-1,j}, \phi_{i+1,j}, \phi_{i,j+1}, \phi_{i,j-1}) \tag{3.7}$$

$$\phi_{max} = max(\phi_{i-1,j}, \phi_{i+1,j}, \phi_{i,j+1}, \phi_{i,j-1}) \tag{3.8}$$

and the normalized values

$$\tilde{\phi}_c = \frac{\phi_{i,j} - \phi_{min}}{\phi_{max} - \phi_{min}} \tag{3.9}$$

$$\tilde{\phi}_2^{(k)} = \frac{\phi_2^{(k)} - \phi_{min}}{\phi_{max} - \phi_{min}} \tag{3.10}$$

$$\tilde{\phi}_1^{(k)} = \frac{\phi_1^{(k)} - \phi_{min}}{\phi_{max} - \phi_{min}}. \tag{3.11}$$

For each *outflow* face (*i.e.*, any face for which (i,j) is the upwind cell) the normalized scalar value of the face $\tilde{\phi}^{(k)}$ is chosen by the following algorithm.

$$\text{if} \quad (\tilde{\phi}_c \leq 0) \quad\quad \text{then} \quad \tilde{\phi}^{(k)} = \tilde{\phi}_1^{(k)}$$

$$\text{elseif} \quad (0 \leq \tilde{\phi}_c \leq \Delta) \quad \text{then} \quad \tilde{\phi}^{(k)} = (\frac{\Delta - \tilde{\phi}_c}{\Delta})\tilde{\phi}_1^{(k)} + (\frac{\tilde{\phi}_c}{\Delta})\tilde{\phi}_2^{(k)}$$

$$\text{elseif} \quad (\Delta \leq \tilde{\phi}_c \leq 1 - \Delta) \quad \text{then} \quad \tilde{\phi}^{(k)} = \tilde{\phi}_2^{(k)}.$$

$$\text{elseif} \quad ((1 - \Delta) \leq \tilde{\phi}_c \leq 1) \quad \text{then}$$

$$\tilde{\phi}^{(k)} = (\frac{\tilde{\phi}_c + \Delta - 1}{\Delta})\tilde{\phi}_1^{(k)} + (\frac{1 - \tilde{\phi}_c}{\Delta})\tilde{\phi}_2^{(k)}$$

$$\text{if} \quad \tilde{\phi}_c > 1 \quad\quad \text{then} \quad \tilde{\phi}^{(k)} = \tilde{\phi}_1^{(k)}$$

This algorithm above simply chooses the face value $\tilde{\phi}^{(k)}$ to be $\tilde{\phi}_1^{(k)}$ if $\tilde{\phi}_c > 1$ or $\tilde{\phi}_c < 0$, and equal to $\tilde{\phi}_2^{(k)}$ if $\Delta \leq \tilde{\phi}_c \leq (1 - \Delta)$, and in the remaining cases chooses $\tilde{\phi}^{(k)}$ equal to a weighted average of the first-order and second-order estimate. The value of Δ obviously has to be less than 0.5. The unnormalized value of $\phi^{(k)}$ can be recovered

from the normalized value by

$$\phi^{(k)} = \tilde{\phi}^{(k)} \left(\phi_{max} - \phi_{min} \right) + \phi_{min}.$$

The sensitivity of the FLOCV to the switching parameter Δ is analysed below for test cases with discontinuities, and others with smooth gradients and extrema. Smaller values of Δ are preferable for *smooth* solutions because this means that the scheme will rarely use first-order upwinding. However, in the presence of shock-like discontinuities, large Δ values are more likely to yield bounded solutions. So the value of Δ should be chosen optimally to trade-off the opposing requirement of accuracy and boundedness. While we cannot expect a given value of Δ to be universally optimum in all situations, our experience has led us to conclude that $\Delta \equiv 0.2$–0.3 would generally work well. The details of the numerical experiments are given in the results section.

3.4 Results

Three steady-state and two time-dependent pure convection problems are solved here and the efficacy of FLOCV in controlling unphysical oscillations near the regions of steep gradients is tested. In all cases the governing equation is (3.1) or its steady-state counterpart, with $\Gamma=0$. The simple explicit temporal scheme is used to solve the time-dependent convection problems (Test Problems 3 and 4).

The performance evaluation is based on the following parameters : (i) the maximum field value, (ii) the minimum field value and (iii) the RMS error. The predicted maximum and minimum values show the effectiveness of the scheme in preserving boundedness, whereas the RMS error is a measure of the overall performance of the scheme. The RMS error is defined as :

$$\mathrm{RMS} = \left[\frac{\sum_{\mathrm{Ntotal}} (\phi^{\mathrm{NUMERICAL}} - \phi^{\mathrm{EXACT}})^2}{\mathrm{Ntotal}} \right]^{1/2} \tag{3.12}$$

where $_{\mathrm{Ntotal}}$ denotes total number of interior grid points.

3.4.1   Test Problem 1

This model problem is same as test problem 4 of chapter 2 (see Figure 2.15). A regular 21×21 grid is used to discretize the solution domain. The other input parameters are $\Gamma = 0$ and the various flow angles corresponding to $y_c = 0$, 1.0, 2.0, 3.0, 4.0, and 5.0. The sensitivity of the FLOCV to the switching parameter (Δ) is studied and results are presented below in the tabular form for different values of Δ at different flow angles. The RMS error along the *mid-plane* ($x=5.0$), the *global* maximum and minimum values of ϕ are shown in Tables 3.1 and 3.2. It can be observed that, except for the $y_c = 0$ case which has very low error, as Δ increases, (a) the RMS errors decrease and then increase, and (b) the levels of undershoots and overshoots decrease. The initial decrease of RMS error is due to the removal of under/overshoots and the RMS error then increases due to fall of accuracy as Δ approaches 0.5. The optimum choice of Δ seems to be between 0.2–0.3 (this conclusion is reinforced by other results below).

Table 3.1: Sensitivity of the FLOCV to switching parameter (Δ) for Test Problem 1

Δ	$Y_c=0$			$Y_c=1.0$			$Y_c=2$		
	Max.	Min.	RMS	Max.	Min.	RMS	Max.	Min	RMS
0.1	1.0	0.0	3.528e-5	1.0	-3.822e-2	0.054	1.0	-3.822e-2	0.0700
0.2	1.0	0.0	5.954e-5	1.0	-2.276e-2	0.0495	1.0	-2.045e-2	0.0691
0.3	1.0	0.0	6.734e-5	1.0	-1.484e-3	0.0527	1.0	-1.794e-8	0.0654
0.4	1.0	0.0	7.596e-5	1.0	0.0	0.0605	1.0	0.0	0.0732

Table 3.2: Sensitivity of the FLOCV to switching parameter (Δ) for Test Problem 1

Δ	$Y_c=3$			$Y_c=4.0$			$Y_c=5$		
	Max.	Min.	RMS	Max.	Min.	RMS	Max.	Min	RMS
0.1	1.0	-3.09e-2	0.075	1.0006	-2.213e-2	0.076	1.009	-9.226e-3	0.0756
0.2	1.0	-9.79e-3	0.073	1.0	-7.405e-6	0.075	1.0	0.0	0.0753
0.3	1.0	0.0	0.071	1.0	0.0	0.072	1.0	0.0	0.0725
0.4	1.0	0.0	0.079	1.0	0.0	0.081	1.0	0.0	0.0844

A comparison of the FLOCV with other schemes is also presented. The results at the

mid plane (x=5.0) are shown in Figures 3.1-3.6 for different flow angles. The results for the conventional upwind scheme, OCV and FLOCV (with Δ=0.3) are shown in the figures. As expected, the conventional upwind scheme does not show any overshoots or undershoots, gives the exact solution when flow is aligned to the grid lines (Y_c=5), but is highly diffusive when the flow is oblique to grid lines. The unbounded OCV scheme shows overshoots and undershoots near the sharp gradient. It can be seen that the FLOCV removes the oscillations associated with the unbounded OCV scheme quite effectively with Δ chosen as 0.3.

The effect of grid irregularity on the solution accuracy of FLOCV is demonstrated by solving the above problem on a distorted grid. The interior grid points are randomly perturbed from their original uniform positions by 5, 10, and 20 per cent of the average grid distance (i.e. each interior grid point is shifted from the uniform grid position by Δx_s, Δy_s in the x and y directions, where Δx_s and Δy_s are uniformly distributed random numbers lying between $\pm(\%distortion) \times \Delta x/100$ and $\pm(\%distortion) \times \Delta y/100$. A typical grid layout with 20 per cent distortion is shown in Figure 3.7. The results are presented in Tables 3.3-3.8 for two grid levels of 21 $\times$ 21 and 41 $\times$ 41 at a flow angle corresponding to Y_c=0 to 3.

The results show that while the RMS errors do increase with increasing grid distortions, the overshoots and undershoots are effectively contained by FLOCV with Δ=0.3 (at the expense of a slight degradation of accuracy).
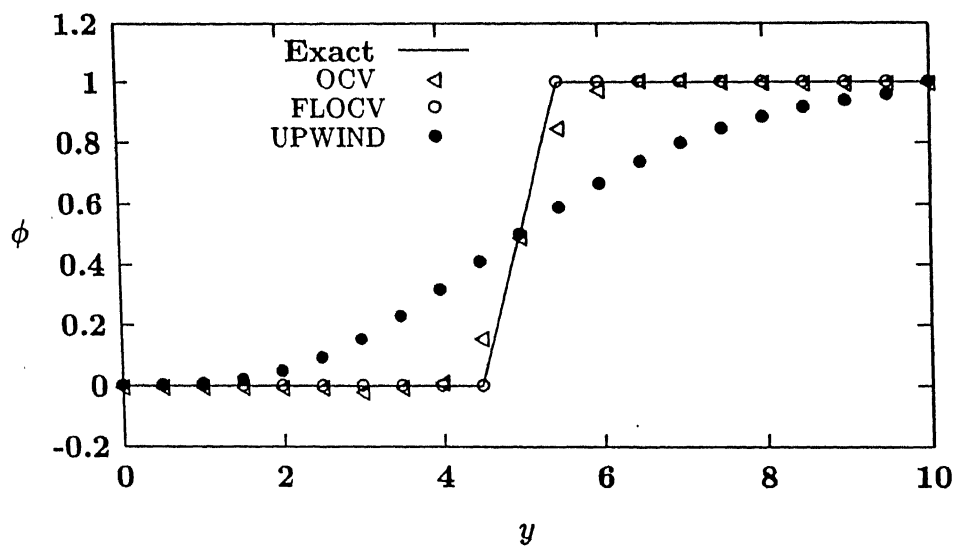
Figure 3.1: Results for test problem 1, $y_c=0$
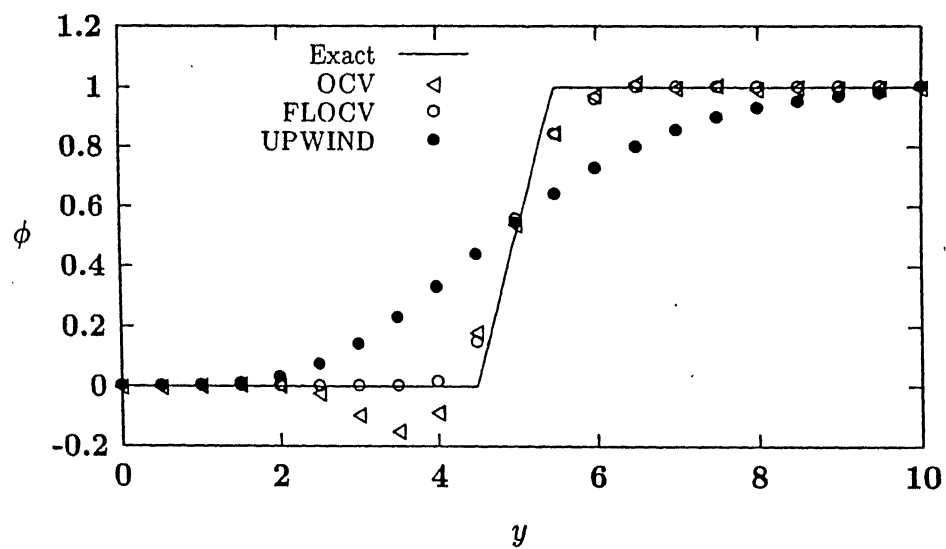
Figure 3.2: Results for test problem 1, $y_c=1$
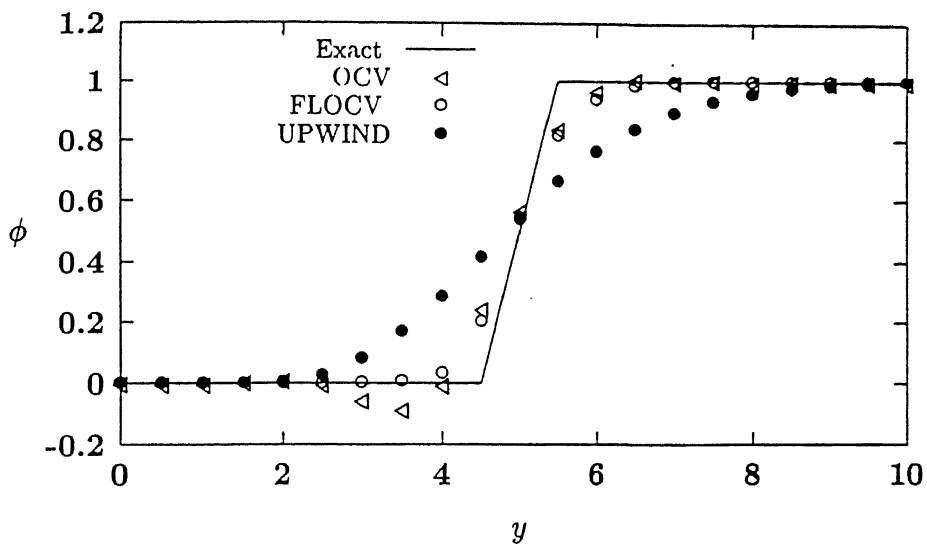
Figure 3.3: Results for test problem 1, y_c=2
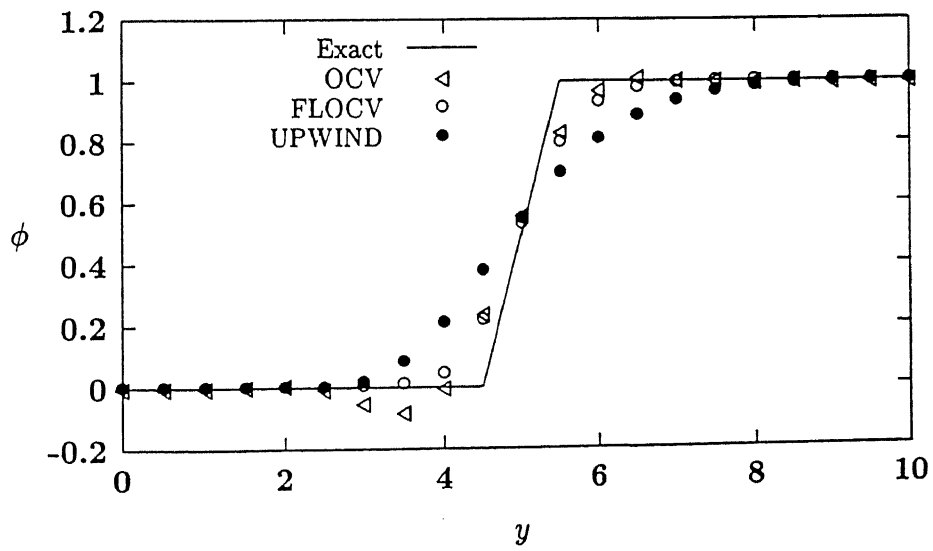
Figure 3.4: Results for test problem 1, y_c=3
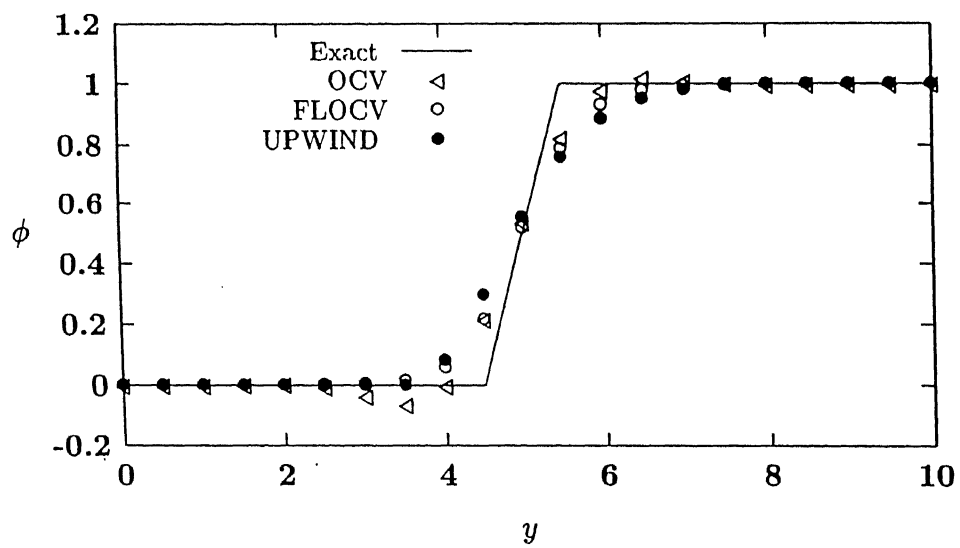
Figure 3.5: Results for test problem 1, $y_c=4$
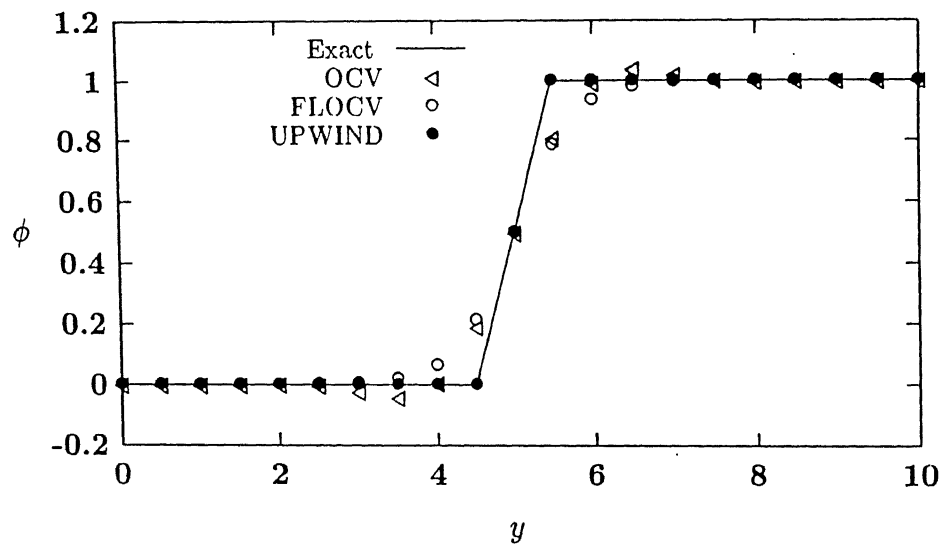
Figure 3.6: Results for test problem 1, $y_c=5$
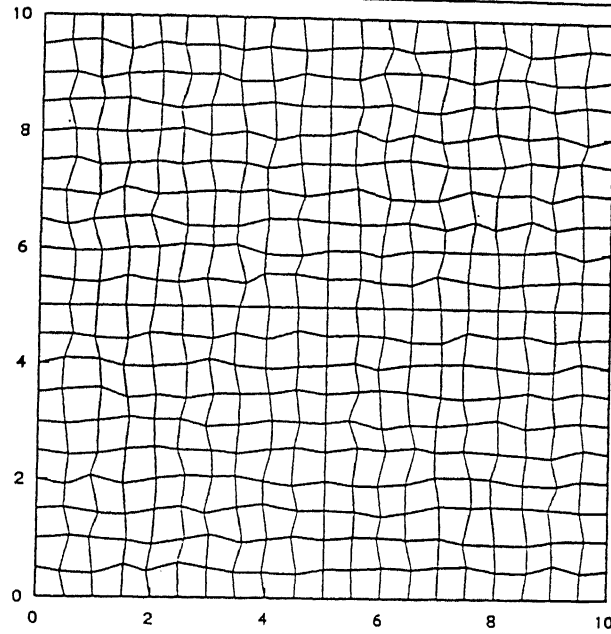
Figure 3.7: Distorted (20 per cent) grid for test problem 1

Table 3.3: Effect of grid distortion (in per cent) on the performance of the **OCV** for Test Problem 1 corresponding to $Y_c=1.0$

Grid size	21×21			41×41		
Distortion	Max.	Min.	RMS	Max.	Min.	RMS
0 per cent	1.03556	-0.21058	0.07123	1.01895	-0.14503	0.05880
5 per cent	1.03676	-.21600	0.07201	1.01951	-0.14342	0.05991
10 per cent	1.03921	-0.21941	0.07312	1.02025	-0.13910	0.06084
20 per cent	1.04412	-0.22277	0.07771	1.02420	-0.12263	0.06441

Table 3.4: Effect of grid distortion (in per cent) on the performance of the **FLOCV** with $\Delta=0.3$ for Test Problem 1 corresponding to $Y_c=1.0$

Grid size	21×21			41×41		
Distortion	Max.	Min.	RMS	Max.	Min.	RMS
0 per cent	1.0	-1.484e-3	0.05274	1.0	-1.4377e-3	0.04711
5 per cent	1.0	-2.1883e-3	0.05311	1.0	-8.2343e-6	0.05041
10 per cent	1.0007	-2.243e-3	0.05407	1.0	-3.1666e-6	0.05438
20 per cent	1.00121	-4.1674e-3	0.05908	1.0005	0.0	0.05999

Table 3.5: Effect of grid distortion (in per cent) on the performance of the **OCV** for Test Problem 1 corresponding to $Y_c=3.0$

Distortion	21×21			41×41		
	Max.	Min.	RMS	Max.	Min.	RMS
0 per cent	1.01750	-0.13531	0.07205	1.01749	-0.15320	0.06440
5 per cent	1.02516	-0.13125	0.07267	1.01644	-0.15039	0.06516
10 per cent	1.03280	-0.12669	0.07350	1.01800	-0.14684	0.06585
20 per cent	1.05102	-0.11633	0.07571	1.02063	-0.13746	0.06702

Table 3.6: Effect of grid distortion (in per cent) on the performance of the **FLOCV** with $\Delta=0.3$ for Test Problem 1 corresponding to $Y_c=3.0$

Distortion	21×21			41×41		
	Max.	Min.	RMS	Max.	Min.	RMS
0 per cent	1.0	0.0	0.07144	1.0	0.0	0.05958
5 per cent	1.0	0.0	0.07275	1.0	0.0	0.06023
10 per cent	1.0	0.0	0.07450	1.0	0.0	0.06260
20 per cent	1.0	0.0	0.07861	1.0	0.0	0.07005

Table 3.7: Effect of grid distortion (in per cent) on the performance of the **OCV** for Test Problem 1 corresponding to $Y_c=5.0$

Grid size	21×21			41×41		
Distortion	Max.	Min.	RMS	Max.	Min.	RMS
0 per cent	1.05793	-0.05793	0.06350	1.06610	-0.06610	0.05563
5 per cent	1.05716	-0.05813	0.06149	1.06790	-0.06660	0.05573
10 per cent	1.05630	-0.05811	0.06353	1.06960	-0.06678	0.05584
20 per cent	1.05454	-0.05806	0.06374	1.07272	-0.06616	0.05614

Table 3.8: Effect of grid distortion (in per cent) on the performance of the **FLOCV** with $\Delta=0.3$ for Test Problem 1 corresponding to $Y_c=5.0$

Grid size	21×21			41×41		
Distortion	Max.	Min.	RMS	Max.	Min.	RMS
0 per cent	1.0	0.0	0.07253	1.0	0.0	0.05933
5 per cent	1.0	0.0	0.07283	1.0	0.0	0.05955
10 per cent	1.0	0.0	0.07407	1.0	0.0	0.06083
20 per cent	1.0	0.0	0.07602	1.0	0.0	0.06585

3.4.2   Test Problem 2

The computational domain is shown in Figure 3.8. The velocity components are $u = y$ and $v = -x$. Again the problem is a purely convective one ($\Gamma=0$), so any scalar profile specified along OA in Figure 3.8 should be swept unchanged along the streamlines and reproduced unchanged at OB, OC and OD (after going through a 90°, 180°, 270° turn, respectively). A scalar profile used by Smith and Hutton[104] is specified along OA :

$$\phi = 1 + \tanh[10(2x + 1)] \quad y = 0, \ -1 \leq x \leq 0$$

The boundary condition is given by

$$\phi = 1 - \tanh(10) \quad \begin{aligned} x &= -1, \ -1 \leq y \leq 1 \\ y &= -1, -1 \leq x \leq 1 \\ y &= 1, \ -1 \leq x \leq 1 \\ x &= 1, \ -1 \leq y \leq 1 \end{aligned}$$

Two numerical solutions are obtained using 41×41 and 81×81 grid points, respectively. The computed profiles after 90, 180 and 270 degree rotations are shown in Figures 3.9-3.14 for the conventional upwind, OCV, and FLOCV ($\Delta=0.3$) schemes. The results for the conventional upwind scheme are very diffusive and the magnitude of numerical diffusion increases as the angle of rotation increases. It can be seen in Figures 3.9-3.11 that the OCV allows small overshoots and undershoots; however, these reduce in magnitude as the mesh is refined, as shown in Figures 3.12-3.14. When FLOCV is used, it removes the overshoots and undershoots (Figures 3.9-3.11) while introducing very little numerical diffusion. The results for the refined mesh, shown in Figures 3.12-3.14, are in close agreement with the exact solution.

This test problem is also solved on distorted grids. The overshoots and undershoots are shown in Table 3.9 to be effectively contained by FLOCV. The level of RMS error for distortion up to 5 per cent is approximately same as that on a uniform grid.

Figure 3.8: Schematic of test problem 3
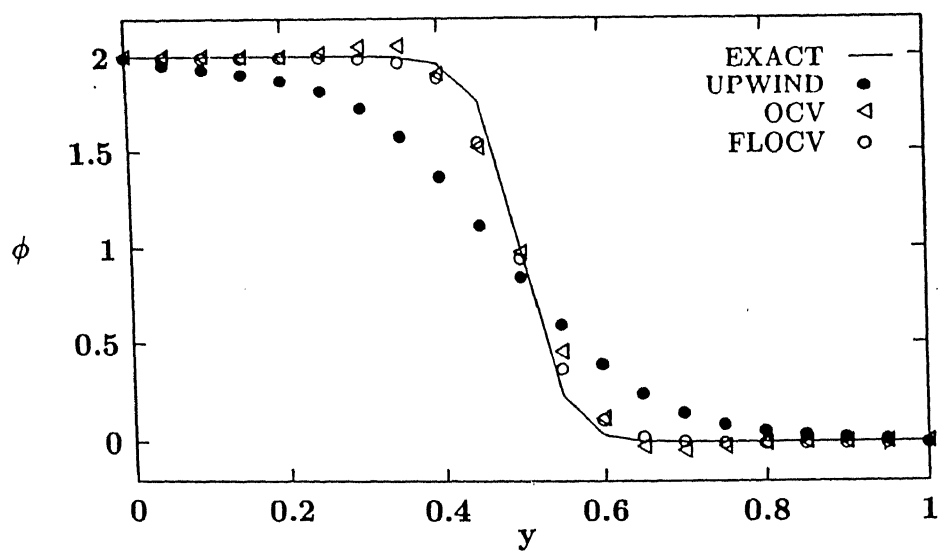
Figure 3.9: Results for test problem 2 along OB ($N=41$)
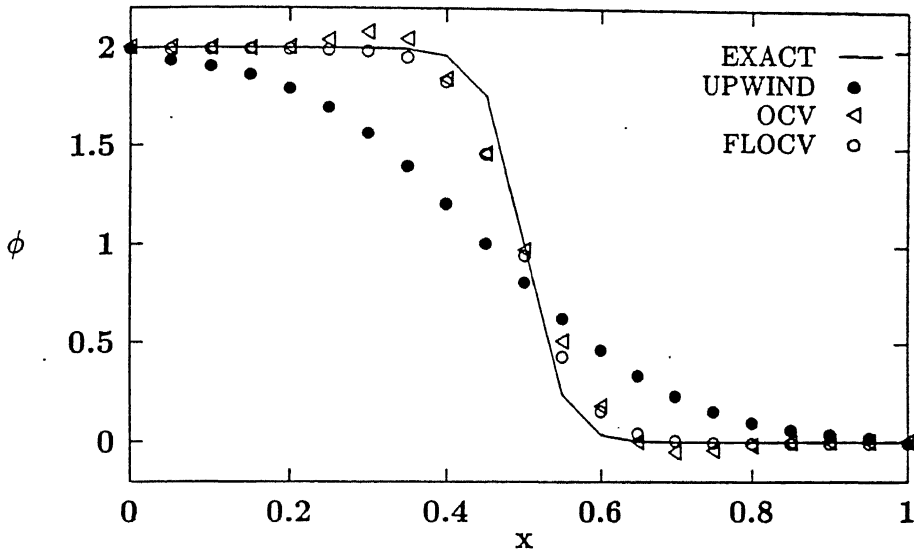
Figure 3.10: Results for test problem 2 along OC (N=41)
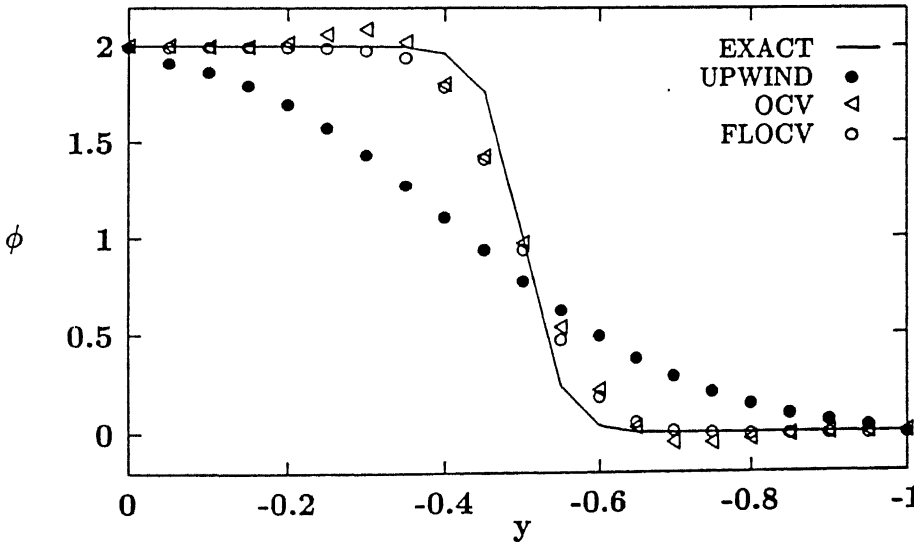
Figure 3.11: Results for test problem 2 along OD (N=41)
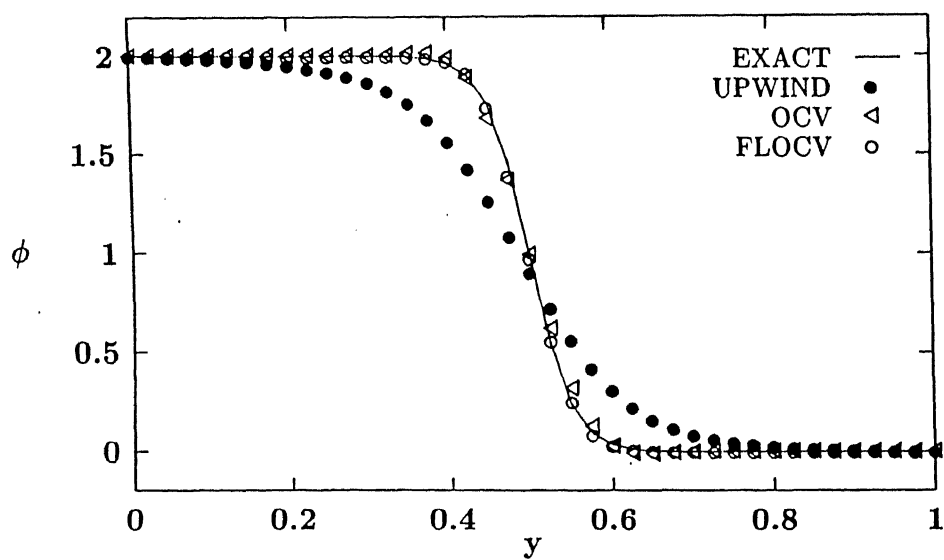
Figure 3.12: Results for test problem 2 along OB (N=81)
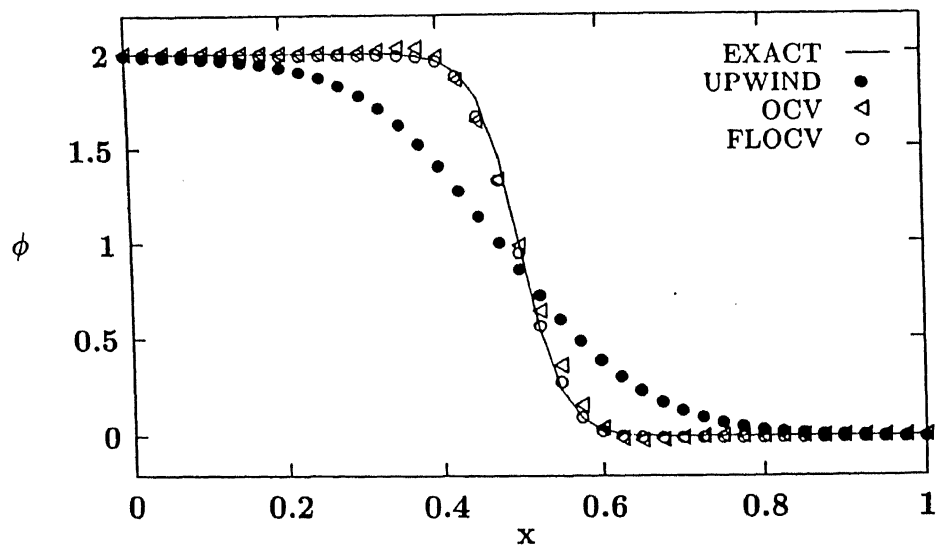
Figure 3.13: Results for test problem 2 along OC (N=81)
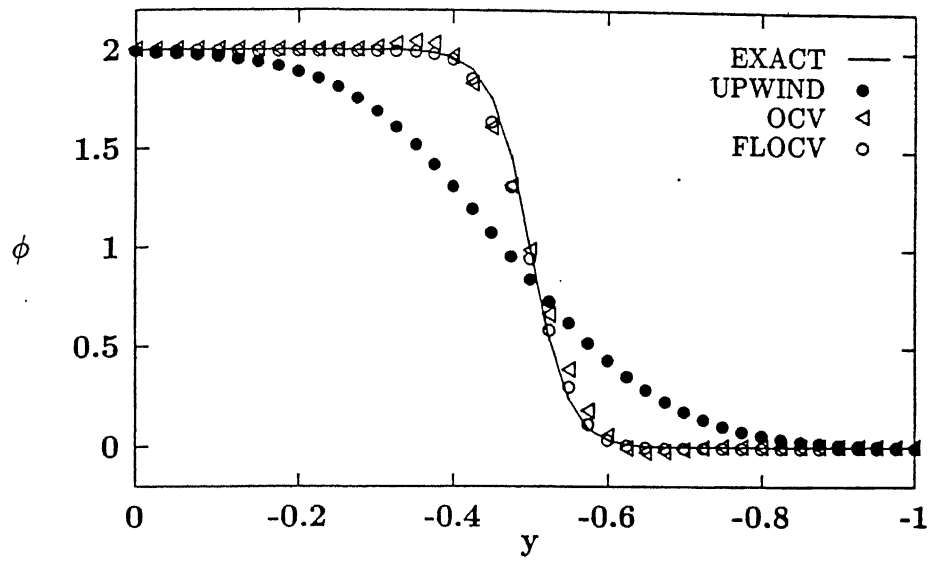
Figure 3.14: Results for test problem 2 along OD (N=81)

Table 3.9: Effect of grid distortion (in per cent) on the performance of the FLOCV with Δ=0.3 for Test Problem 2

Grid size	41×41			81×81		
Distortion	Max.	Min.	RMS (along OD)	Max.	Min.	RMS (along OD)
0 per cent	1.9999	0.0	0.1123	1.9999	0.0	0.0326
5 per cent	2.0005	0.0	0.1142	2.0002	0.0	0.0347
10 per cent	2.0008	0.0	0.1350	2.0003	0.0	0.0423
20 per cent	2.0016	-.2e-11	0.1853	2.0005	0.0	0.0937

3.4.3   Test Problem 3

This problem is that of an advancing front in a unidirectional flow and is included to demonstrate the capability of the proposed flux limiter in removing overshoots and undershoots in unsteady computations.

A schematic of the problem is shown in Figure 3.15. The scalar in the domain is initially zero everywhere. A step input is specified at the the inlet for $t \geq 0$. The boundary conditions at the other boundaries are homogeneous Neumann conditions. The components of velocity u and v are 0.5 and 0.0, respectively. The diffusion coefficient (Γ) is zero, *i.e.*, the flow is purely convective. Therefore any profile specified at the inlet is advected downstream without any change and can be used as a exact solution at the front position at any point in time. The computational domain is divided into a 65×65 uniform mesh. Two different time steps, $\Delta t=$ 48 and 96 units, corresponding to Courant numbers ($\equiv \frac{u\Delta t}{\Delta x}$) 0.12 and 0.24, respectively, are used in the computations. The results are shown (for $t = 9600$) shown in Figures 3.16-3.17 for OCV and FLOCV along with the exact solution. It can be seen that oscillations are removed by FLOCV.

The performance of the flux limiter along with the base scheme can also be judged on the basis of the following three parameters, E_w, the total variation of errors or the waviness, E_t, the total absolute error, and E_s, the spreading index as defined below:

$$E_w = \sum_{i=1}^{mmax} \mid e_{i+1} - e_i \mid \qquad (3.13)$$

where e_i is the error in the solution at grid point i (at j=j_{mid} corresponding to the centerline $y = 0$),

$$E_t = \sum_{i=1}^{mmax} \Delta x_i \mid e_i \mid, \qquad (3.14)$$

and

$$E_s = \sum_{i=1}^{mmax} \Delta x_i \mid x_i - x_f \mid \quad \mid e_i \mid, \qquad (3.15)$$

where x_f is the position of the front at the given time.
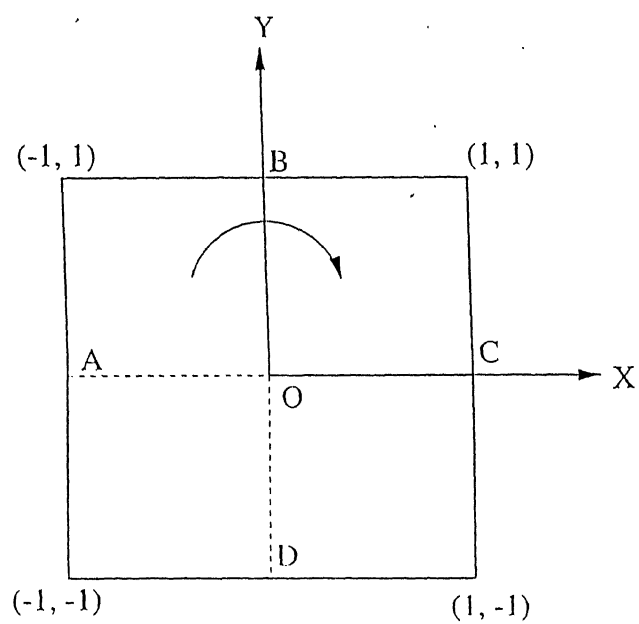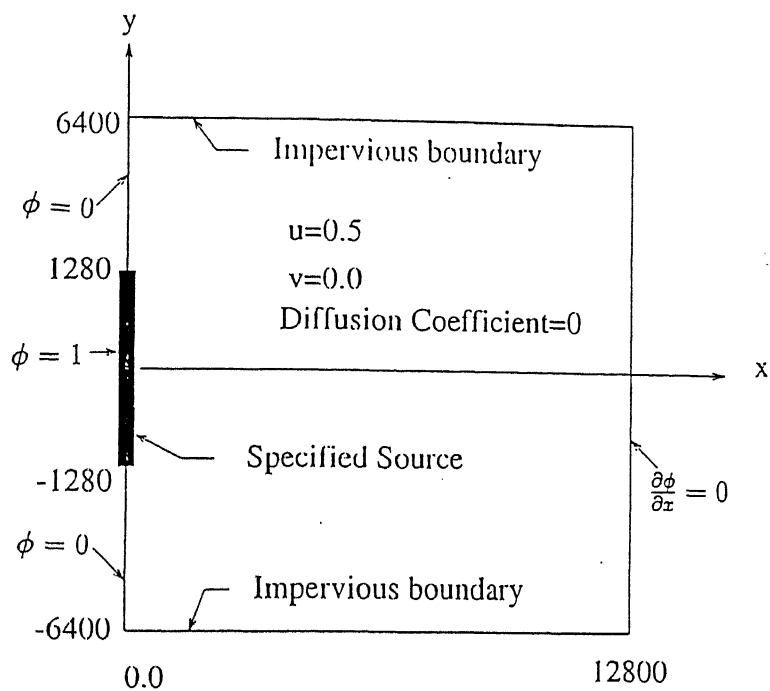
Figure 3.15: Schematic of test problem 3
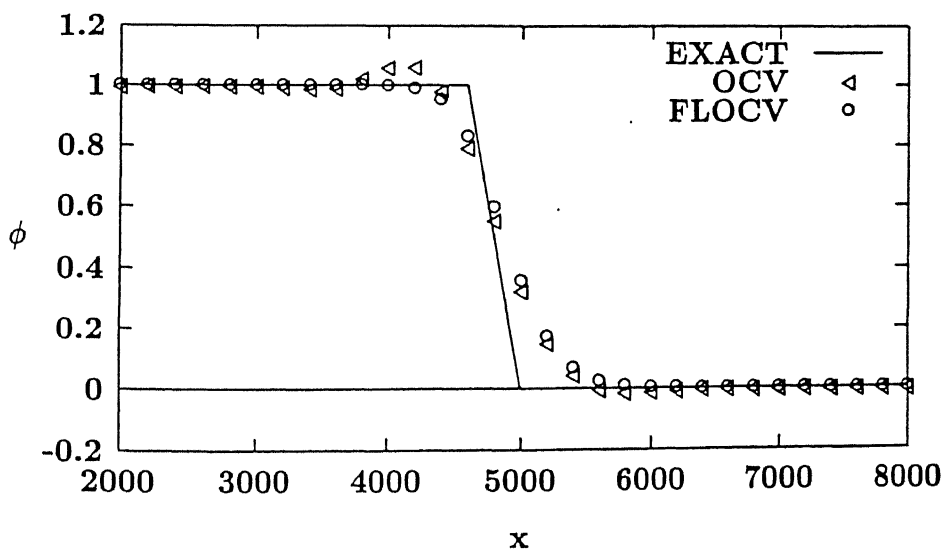
Figure 3.16: Results for test problem 3 (Δt=48)
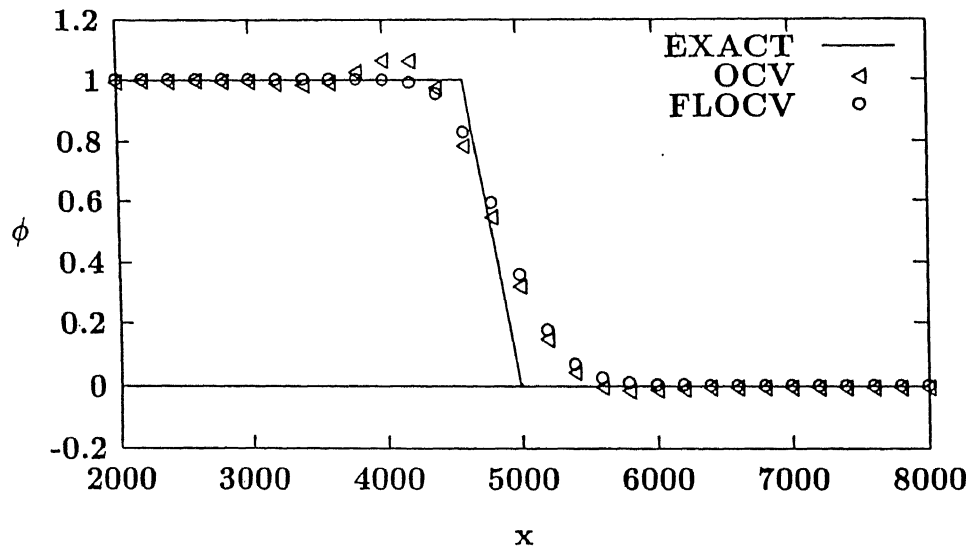
Figure 3.17: Results for test problem 3 (Δt=96)

Table 3.10: Test problem 3 - Total Variation of Errors or the Waviness (E_w)

Δt	OCV	FLOCV
48	1.75	1.52
96	1.76	1.52

Table 3.11: Test problem 3 - Total Absolute Error (E_t)

Δt	OCV	FLOCV
48	281.46	252.95
96	286.64	254.24

Table 3.12: Test problem 3 - Spreading Index (E_s)

Δt	OCV	FLOCV
48	8.78E4	5.49E4
96	9.08E4	5.53E4

The performance parameters are presented in Tables 3.10, 3.11, and 3.12 for the OCV and FLOCV schemes. The flux limiter considerably improves the performance of the original scheme. It smoothens the waviness in the solution, reduces total absolute error, and decreases the numerical spreading.

3.4.4 Test Problem 4

In this test case a square-shaped scalar field is advected diagonally across a square domain. The schematic of the problem is shown in Figure 3.18. The source field (shown dark) has a scalar value of 10 and the rest of the domain has a scalar value of 0. The scalar field was initially centered at a point (-1.5, -1.5) and advected by a uniform velocity field, making 45° with the coordinate lines, to a position (1.5, 1.5). The components of the velocity u, and v are of unit magnitude.
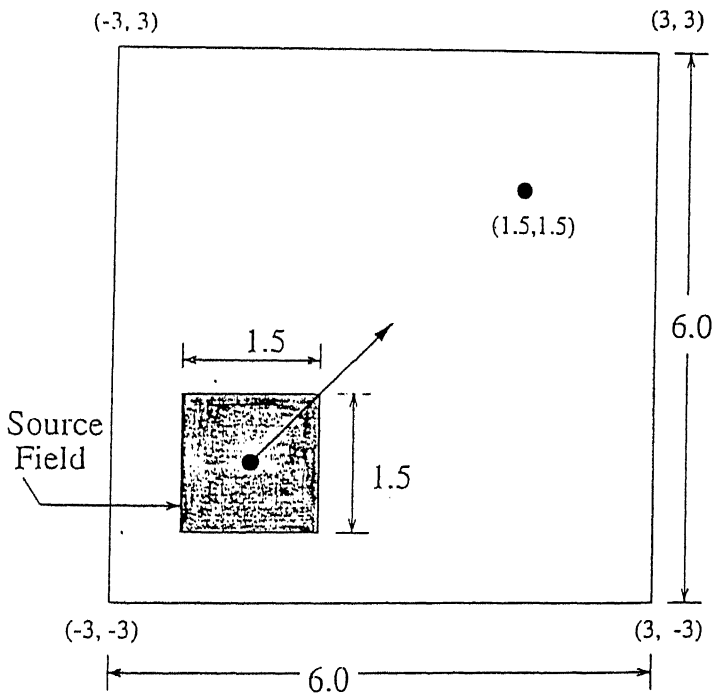
Figure 3.18: Schematic of test problem 4
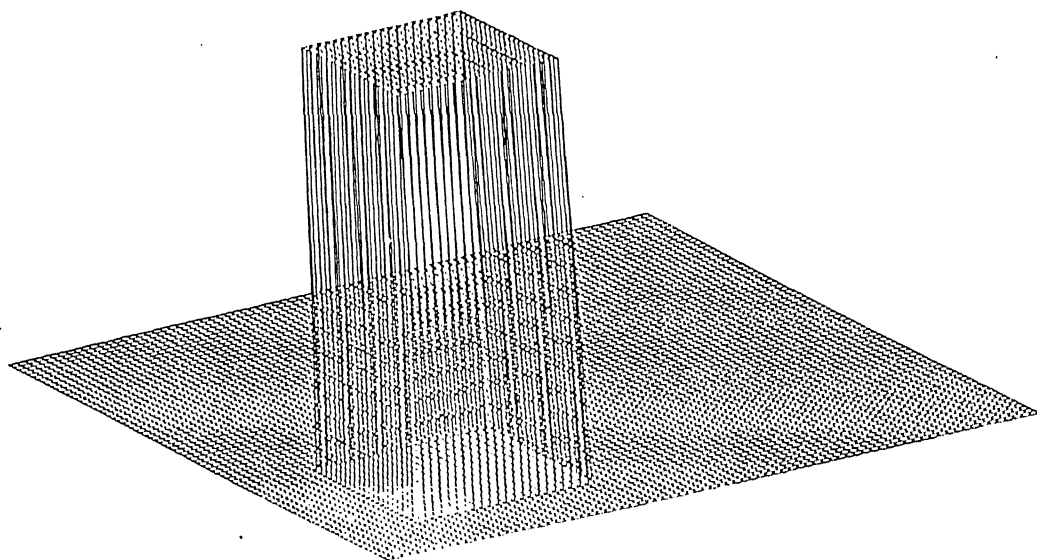
Figure 3.19: The 3-D perspective plot of initial scalar field for test problem 4
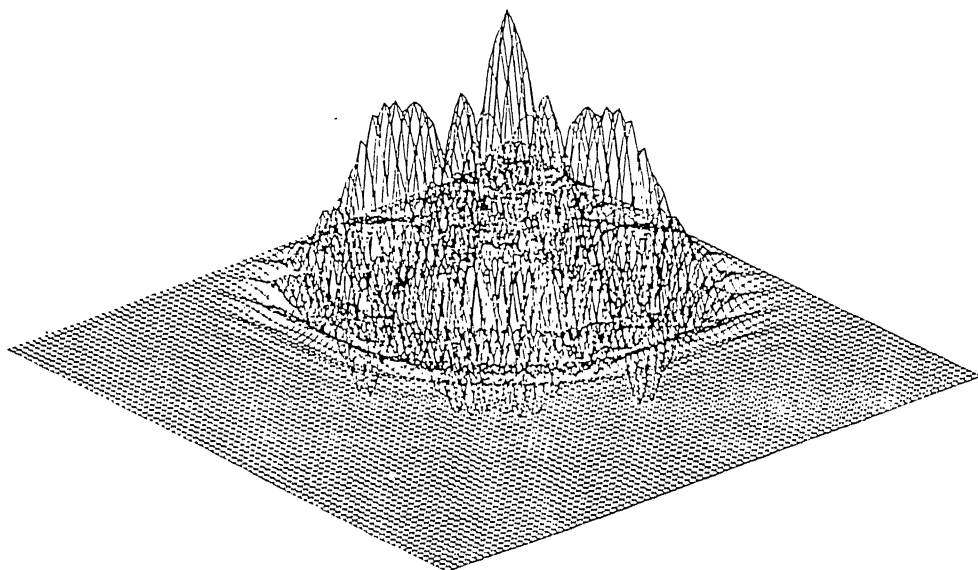
Figure 3.20: The 3-D perspective plot of scalar field predicted by OCV scheme
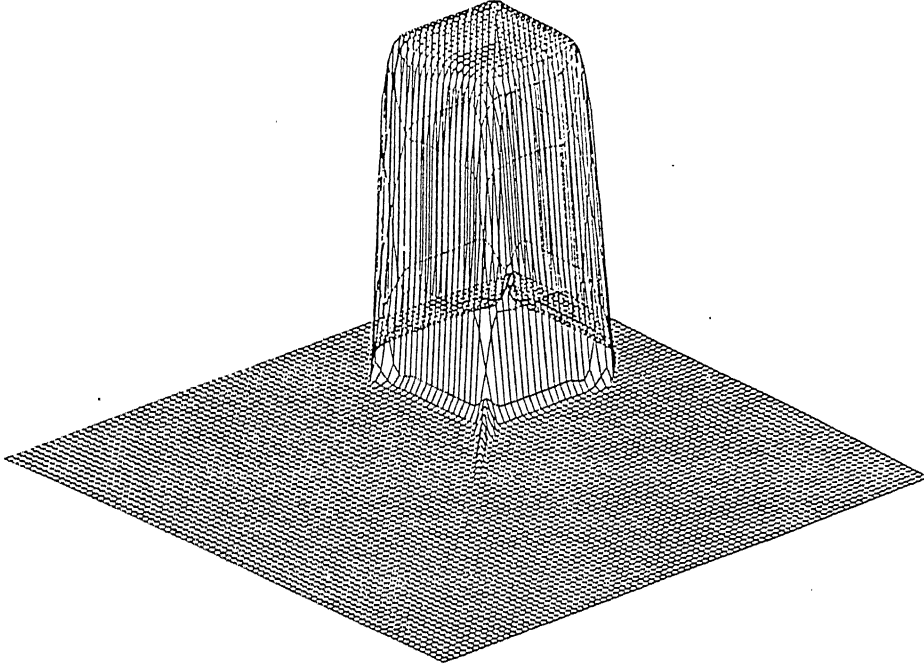
Figure 3.21: The 3-D perspective plot of scalar field predicted by FLOCV scheme

This test is very stringent as many lines of discontinuities exist in the problem. The perspective plots of the scalar field are shown in Figures 3.19-3.21 for the OCV and FLOCV results for a grid of 81 × 81 and a Courant number ($C_n = \frac{u\Delta t}{\Delta x}$) of 0.20. The initial scalar field is shown in Figure 3.19, and the final fields in Figures 3.20 and 3.21, for OCV and FLOCV, respectively. It can be seen that the flux limiter works well and dramatically improves the solution. The *global* RMS error and minimum and maximum of the computed scalar field for the OCV and FLOCV are presented, for two grids of 41 × 41 and 81 × 81 grid points, in Table 3.13 for $C_n = 0.20$. It can be seen that the flux limiter works satisfactorily and considerably improves the results. Tamamidis ans Assanis[105] have compared the performance of various schemes for this problem. These results are shown in Table 3.14. It is clear that the performance of FLOCV compares well with that of MPL[37], MSOU[106], and SHARP[36].

Table 3.13: Results for test problem 4 – Courant number = 0.20

Scheme	Grid 41 × 41			Grid 81 × 81		
	Max.	Min.	RMS	Max.	Min.	RMS
OCV	23.164	-7.658	1.999	41.006	-26.727	4.173
FLOCV	9.992	0.0	0.755	10.0	0.0	0.5312

Table 3.14: Test problem 4 – Results reported by Tamamidis and Assanis[105]

Scheme	40 × 40			80 × 80		
	Max.	Min.	RMS	Max.	Min.	RMS
FOU	6.257	0.0	1.445	8.526	0.0	1.289
SOU	16.356	-3.58	1.268	18.880	-5.575	1.329
QUICK	18.808	-5.88	1.737	35.471	-21.092	3.366
MPL	9.973	0.0	0.936	10.0	0.0	0.717
MSOU	10.0	0.0	0.855	10.0	0.0	0.537
SHARP	10.219	-0.44	0.948	10.892	-1.373	0.652

3.4.5 Test Problem 5

This test was used to show the efficacy of the convection modelling of the OCV method in chapter 2. This test case computes a smooth solution and is selected to demonstrate the effect of flux limiting on the order of accuracy and to show how diffusive and/or over-compressive FLOCV might be. The computational domain and boundary conditions are the same as shown in Figure 3.8. The velocity components are defined as $u = y$ and $v = -x$. Again the problem is purely convective ($\Gamma=0$), so any scalar profile specified along OA in Figure 3.8 should be reproduced unchanged at OD (after going through a 270° turn). The profile specified along OA is the smooth Gaussian distribution $\phi = e^{2|x|} sin^2(\pi x)$.

The computational domain is discretized using $N \times N$ uniform grids with N=41, 51, 61, 71 and 81. The RMS error for the points on OD (*not* for the entire domain) for OCV and FLOCV with different values of Δ at various grid levels are shown in Figure 3.22 on a log-log scale. If the error ϵ is assumed proportional to Δx^m, where m is the order of the method then the values of m are 2.7481, 2.6327, 2.2547 and 1.7788 for OCV and FLOCV with Δ= 0.1, 0.2, 0.3 respectively. It can be seen that FLOCV maintains second-order accuracy upto Δ=0.2 and there is only small deterioration for Δ=0.3.

The computed ϕ profile along OD at grid levels of 41 × 41 and 81 × 81 are shown in Figures 3.23-3.24 for different values of Δ. It can be observed that FLOCV is slightly compressive (i.e. it tends to *square-off* the solution near the peak, especially for Δ=0.3 on the coarse mesh). However, grid refinement improves the solution considerably in Figure 3.24. On both the coarse and fine mesh diffusive errors are small.

The solution of FLOCV for this smooth case was also obtained on the non-uniform orthogonal grid shown in Figure 3.25. (Each quadrant of this grid is a Gauss-Lobatto Chebyshev grid of size $M \times M$ resulting in a composite $N \times N$ grid of $N = 2M + 1$). A typical grid layout with 41 × 41 grid points is shown in Figure 3.25. The RMS error along OD is presented in the Figure 3.26 for OCV and FLOCV (with Δ=0.1, 0.2 and 0.3). Once again if we assume error $\epsilon \propto (1/N)^m$, the exponents m are 2.591, 2.270, 2.182, 1.707 for OCV and FLOCV with Δ=0.1, 0.2, 0.3, respectively. It is
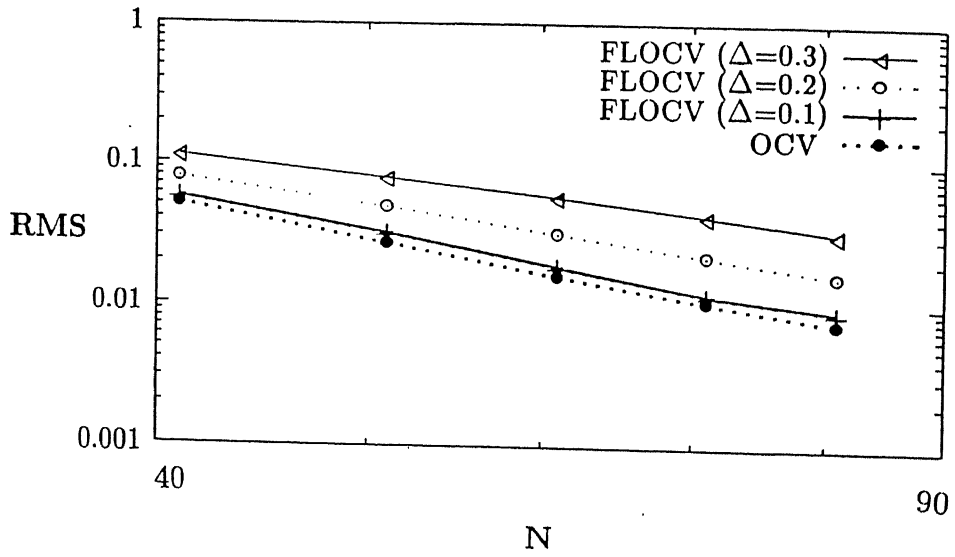
Figure 3.22: RMS versus N for test problem 5 along OD (Uniform grid)
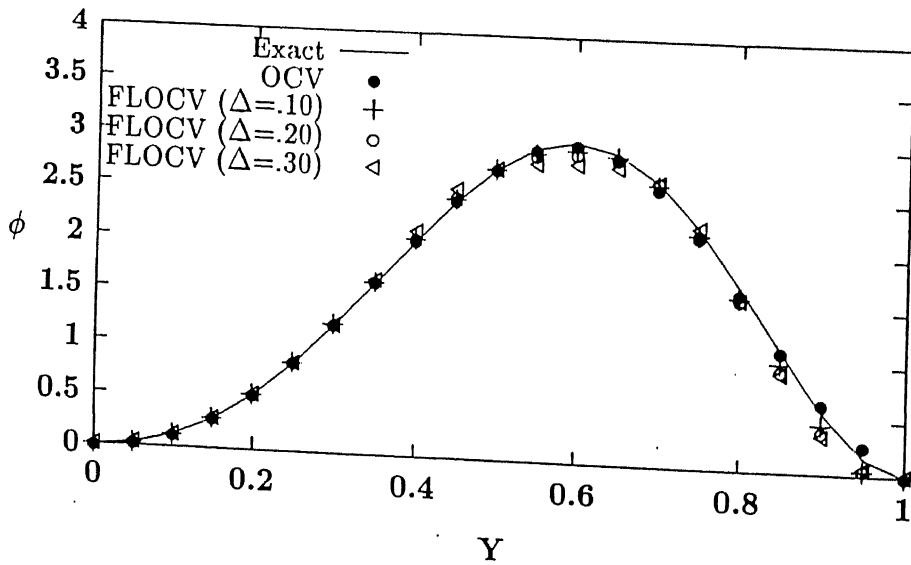
Figure 3.23: Computed profile along OD on 41×41 grid
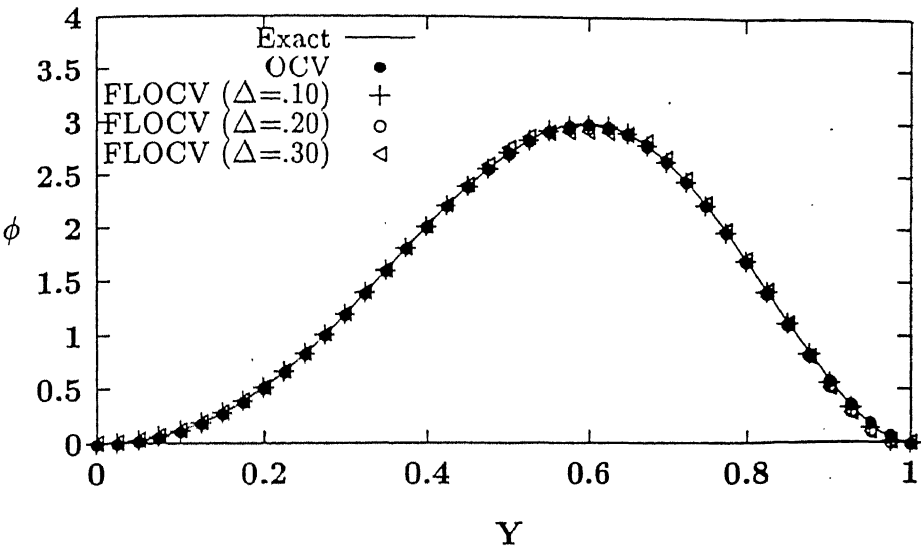
Figure 3.24: Computed profile along OD on 81 × 81 grid
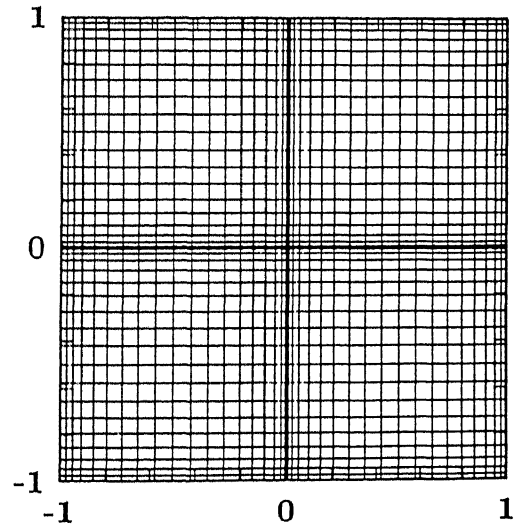
Figure 3.25: Non-uniform Cartesian grid
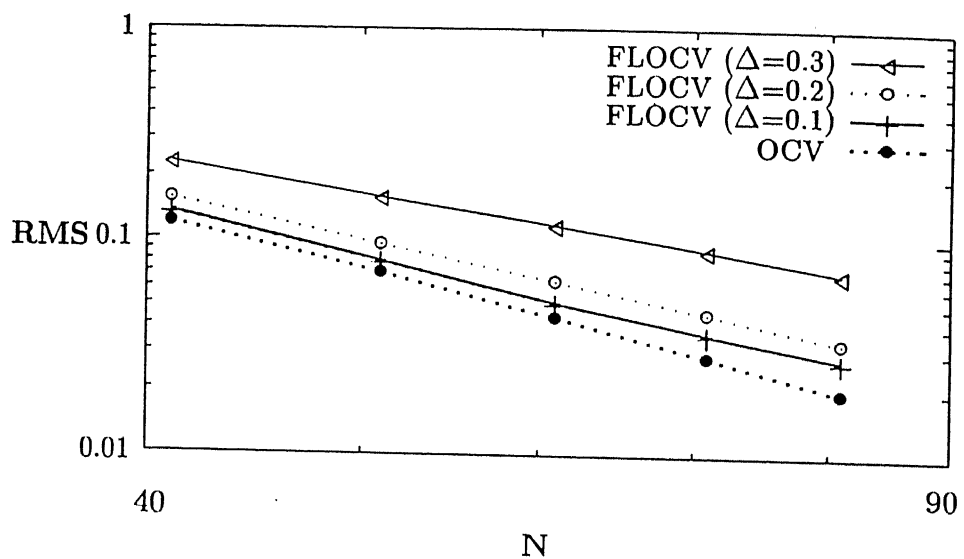
Figure 3.26: RMS versus N for test problem 5 along OD (Non-uniform grid)
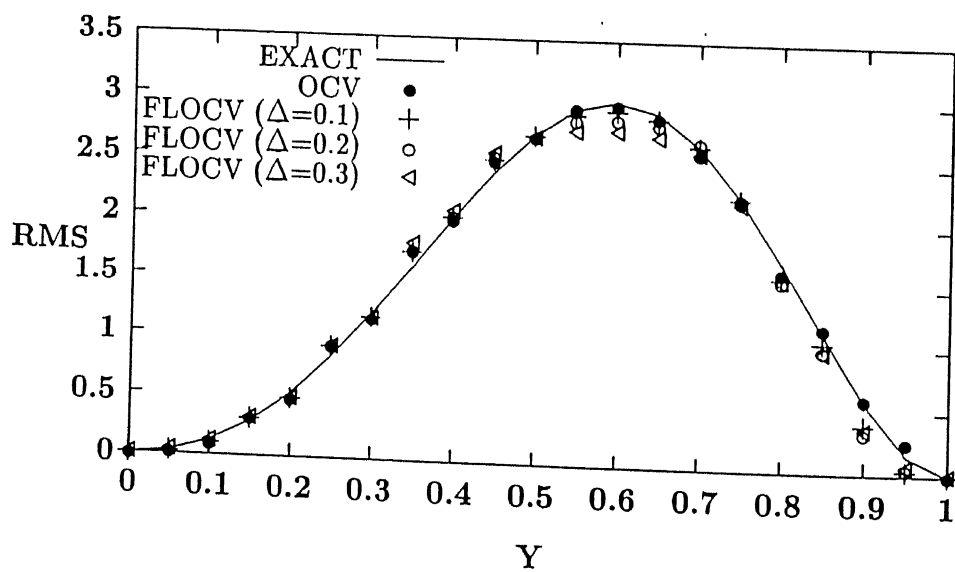
Figure 3.27: Computed profile along OD on 20 per cent distorted grid (41×41)
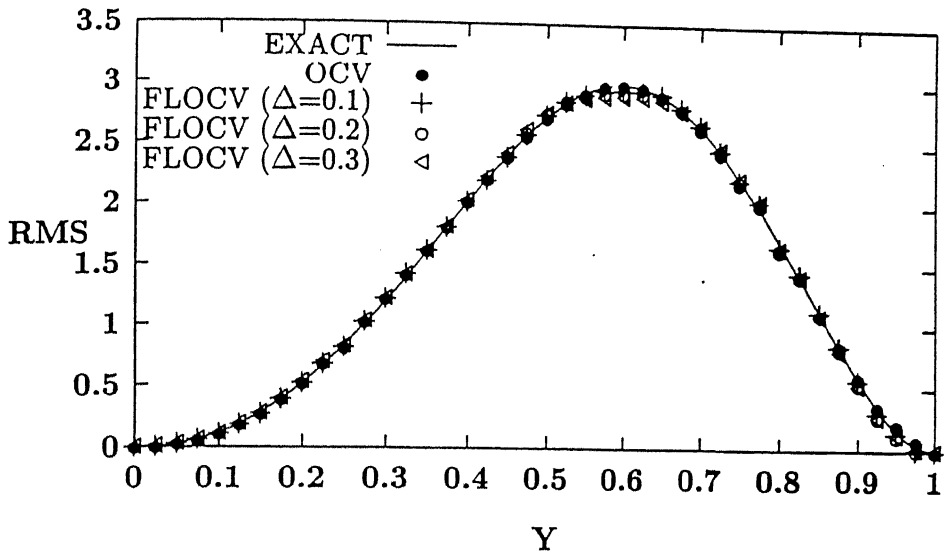
Figure 3.28: Computed profile along OD on 20 per cent distorted grid (81 × 81)

perhaps worthy of mention that the Chebyshev grid used, while smooth, is *highly* non-uniform; for example, with $N = 81$, the ratio of the largest to smallest grid intervals is approximately 25. It can be seen that not only is OCV second order accurate on this non-uniform grid, but also that FLOCV retains this order of accuracy till at least $\Delta=0.2$.

Next, we take up the case of *non-uniform, non-orthogonal* grids. The grid is distorted from the uniform grid by the same means employed in test problem 1. The effect of grid distortion on the solutions of OCV and FLOCV are shown in Tables 3.15 to 3.18. The table shows RMS errors along OD (270 ° turn) from the initial Gaussian profile specified at OA. It can be seen that FLOCV retains second-order accuracy up to 10 per cent grid distortion for $\Delta \leq 0.2$. The deterioration in the order-of -accuracy of the FLOCV is not very much even on the highly (20 per cent) distorted grid. The computed profiles along OD are also shown in Figures 3.27- 3.28 for the 20 per cent distorted grid.

Table 3.15: Test problem 5 – The RMS error along OD on 0 per cent distorted grid

Grid	OCV	FLOCV ($\Delta=0.1$)	FLOCV ($\Delta=0.2$)	FLOCV ($\Delta=0.3$)
41 × 41	0.05019	0.05692	0.07760	0.11221
51 × 51	0.02657	0.03154	0.04825	0.07899
61 × 61	0.01585	0.01858	0.03141	0.05711
71 × 71	0.01048	0.01190	0.02199	0.04195
81 × 81	0.00747	0.00917	0.01626	0.03178

Table 3.16: Test problem 5 – The RMS error along OD on 5 per cent distorted grid

Grid	OCV	FLOCV ($\Delta=0.1$)	FLOCV ($\Delta=0.2$)	FLOCV ($\Delta=0.3$)
41 × 41	0.05111	0.06002	0.07986	0.11509
51 × 51	0.02669	0.03214	0.04911	0.07944
61 × 61	0.01844	0.02101	0.03368	0.05938
71 × 71	0.01168	0.01339	0.02400	0.04349
81 × 81	0.00855	0.00995	0.01682	0.03288

Table 3.17: Test problem 5 – The RMS error along OD on 10 per cent distorted grid

Grid	OCV	FLOCV ($\Delta=0.1$)	FLOCV ($\Delta=0.2$)	FLOCV ($\Delta=0.3$)
41 × 41	0.05443	0.06488	0.08592	0.11868
51 × 51	0.02933	0.03619	0.05306	0.08163
61 × 61	0.02275	0.02687	0.03940	0.06421
71 × 71	0.01406	0.01656	0.02782	0.04667
81 × 81	0.01160	0.01275	0.01996	0.03544

Table 3.18: Test problem 5 – The RMS error along OD on 20 per cent distorted grid

Grid	OCV	FLOCV ($\Delta=0.1$)	FLOCV ($\Delta=0.2$)	FLOCV ($\Delta=0.3$)
41 × 41	0.06639	0.08012	0.10363	0.13141
51 × 51	0.03964	0.05311	0.06749	0.09256
61 × 61	0.03372	0.04162	0.05423	0.07391
71 × 71	0.02059	0.02617	0.03818	0.05331
81 × 81	0.01844	0.02267	0.02951	0.04088

3.5 CPU-time comparison and Programming considerations

For steady-state cases, the steady convection-diffusion equation is solved implicitly using an approach similar to deferred-correction approach[103]. To compare the CPU time required by the OCV and FLOCV schemes, Test Problem 5 is again used here. The convergence criterion selected for this problem is $| (\phi_{(i,j)}^{(p+1)} - \phi_{(i,j)}^{(p)}) | < 10^{-5}$ where p is an iteration index. The results are tabulated below for OCV and FLOCV (with $\Delta = 0.3$ on uniform Cartesian grids. It can be seen in Table 3.19 that the increase in CPU time for FLOCV is approximately 10–15 per cent compared to OCV. The calculation were done on a DEC-3000 machine.

Table 3.19: CPU-time (in seconds) comparison for Test Problem 5 on uniform Cartesian grid

Grid	OCV	FLOCV (Δ=0.3)
41 × 41	52.137	57.288
51 × 51	104.692	111.764
61 × 61	183.088	203.998
71 × 71	327.558	352.860
81 × 81	525.349	597.575

3.6 Summary

The scheme FLOCV is a flux limited version of the base OCV scheme. FLOCV switches from second to first order interpolation in the presence of extrema. First order interpolation introduces numerical diffusion which damps oscillations. The switching between the different interpolation methods is controlled by a blending parameter Δ.

We have tested the performance of FLOCV under varied and arduous conditions. The cases considered in this study include steep gradients, flow-to-grid skewness and circulating flows on uniform and non-orthogonal structured grids. FLOCV performs well in all the test cases considered. The salient results are :

- For problems with discontinuities, FLOCV is quite effective in removing oscillations associated with the unbounded OCV scheme on both orthogonal and non-orthogonal grids.

- The effect of flux limiting on the accuracy of the base scheme was studied using a smooth Gaussian profile. It was demonstrated that FLOCV is second-order accurate on uniform Cartesian grids for $\Delta \leq 0.2$. FLOCV was also found to retain this order of accuracy even on highly non-uniform Cartesian grids. On mildly non-orthogonal grids (up to 10 percent distortion) FLOCV is second order accurate with $\Delta \leq 0.2$. The deterioration in the order-of-accuracy of the FLOCV is not much on a highly (20 percent) distorted grid.

- FLOCV was also applied to unsteady test cases. It smoothens waviness in the solution and decreases numerical spreading as compared to the OCV scheme.

- The performance of FLOCV was judged with other standard flux limiting schemes in a stringent test problem (Test Problem 4). FLOCV was comparable to the best of the other schemes.

- The sensitivity of the FLOCV to the switching parameter Δ was analysed. The optimum choice of Δ seems to be between 0.2–0.3. (A more precise value can be given only for specific problems).

- FLOCV can be a good choice for 2-D convective-diffusive problems on orthogonal and non-orthogonal structured grids.

Chapter 4

Application of the OCV Method to Solute Transport Problem

4.1 Introduction

Modelling the transport of dissolved solutes in ground water flows of practical interest requires the numerical solution of a transient convection-dispersion equation in two, or more, dimensions. The numerical schemes for these equations need to have sufficient accuracy and also be adaptable to complex geometries, which are common in practical applications of solute transport problems. The accuracy of a scheme is classified by its order of accuracy in time and space. Integrations of first-order accuracy have diffusive errors which tend to spread the solution at sharp fronts. Therefore, second-order integration schemes are preferable for problems with such fronts. However, second-order schemes tend to produce solutions with spurious oscillations at the fronts.

Finite-element methods (FEM) are popular for solving the transport equations (see, *eg*, Huyakorn and Pinder[107]). Finite-difference methods (FDM) though simple and easy to apply in rectangular domains, have difficulty in handling complex geometries. The finite-element methods are algorithmically more complex, but can be applied on irregular geometries. One of the main problems with the conventional Galerkin finite-element formulation is its inability to handle convection dominated flows. Van Genuchten[108], Pinder and Shapiro[109], Heinrich et al.[110], Sun and Yeh[111], Wang et al.[112], Yeh[113], and Westerink and Shea[114] have developed finite-element methods which attempt to

minimize the numerical oscillations in various ways. Yu and Singh[115] have recently developed a modified Galerkin finite-element method for solute transport.

Eulerian-Lagrangian methods have also been developed in recent years for the solution of transport equations. In these methods, the advection-dispersion equation is decomposed into two parts : one modelling pure advection and the other dispersion. The Lagrangian approach is used for the advection part and the Eulerian approach for the dispersion part. These techniques reduce the numerical oscillations, but produce false-diffusion at the front. Yeh[116], Yeh and Chang[117] and Ijiri and Karasaki[118] use various techniques to minimize this diffusion. These schemes achieve a better accuracy, but at the expense of higher computational costs.

Finite-volume techniques offer a viable alternative to the finite-element methods for solving flow and transport problems (see, eg, Peyret and Taylor [119]). They combine the flexibility of handling complex geometries, intrinsic to FEM, with the simplicity of FDM. Putti et al.[120] developed a triangular finite-volume technique for solving the ground water solute transport equation. This method uses a monotone interpolation scheme in order to avoid the numerical oscillations at sharp concentration fronts in advection dominated flows. It is an explicit method in which the computational time step is limited by the CFL condition. Cox and Nishikawa[121] presented a total variation diminishing (TVD) scheme based on rectangular orthogonal elements. This method is also based on an explicit formulation for the time-stepping.

In this chapter we adopt the OCV technique to address the specific problem of multi-dimensional transient solute transport in ground water. The method has the following features:

1. It can handle a variable tensorial diffusivity, common in solute transport.

2. The time-stepping is implicit and unconditionally stable, as solute transport problems often have long integration times for which explicit schemes restricted by CFL stability condition are expensive.

3. The scheme is second-order in space and when combined with Crank-Nicolson time-stepping scheme is second-order in time as well.

4. The method can be used on non-orthogonal geometries and with Dirichlet, Neumann, and Robin boundary condition.

5. It can be used for large range of grid Peclet numbers and Courant numbers and is suitable for both diffusion and convection dominated flows.

This chapter presents the formulation of the scheme and the results of three test cases which demonstrate its capabilities. Certain difficulties in the time-stepping scheme, such as the suppression of spurious oscillations are also addressed.

4.2   Governing Equations

The governing equation for solute transport under saturated ground water flow conditions is given by (Freeze and Cherry[122])

$$R_d \frac{\partial C}{\partial t} = \nabla.(D_h.\nabla C) - \nabla.(\tilde{V}C) - \lambda R_d C \qquad (4.1)$$

where, C is the solute concentration, $\tilde{V}$ is the pore water velocity vector, R_d is the retardation factor, λ is the first-order decay coefficient, D_h is the hydrodynamic dispersion tensor and t is time. The elements of the dispersion tensor D_{xx}, D_{zz}, D_{xz} $(= D_{zx})$ are generally functions of velocity and the molecular diffusion. In the numerical studies attempted here only the case of a homogeneous and isotropic medium under two-dimensional ground water flow with two-dimensional dispersion is considered. However, the formulation below is a general one and can be applied to non-homogeneous, non-isotropic conditions.

4.3   Finite-Volume Formulation

The solution domain is discretized into a structured non-orthogonal grid as in chapter 2 (Figure 2.1). Each control volume is labelled by the index (i, j) of the central node.

On integrating the equation (4.1) over the control volume and applying the Gauss-divergence theorem, we get

$$\int\int R_d \frac{\partial C}{\partial t} dA \;=\; \oint_{cs} \left[\left(D_{xx}\frac{\partial C}{\partial x} + D_{xz}\frac{\partial C}{\partial z} \right) n_x \right] dl$$

$$+ \; \oint_{cs} \left[\left(D_{zx}\frac{\partial C}{\partial x} + D_{zz}\frac{\partial C}{\partial z} \right) n_z \right] dl$$

$$- \; \oint_{cs} C \left(u n_x + w n_z \right) dl - \int\int \lambda R_d C dA \qquad (4.2)$$

where dl is an elemental length on the boundary (control surface) of the control volume, n_x and n_z are the direction cosines of the local outward unit vector on the boundary in x and z directions, respectively, and dA is the elemental area of the control volume. Equation (4.2) can be partially discretized as

$$\frac{R_d A_s}{\Delta t} \left(C^{n+1} - C^n \right) \;=\; \theta \; [DIFF]^{n+1} + (1-\theta)[DIFF]^n \qquad (4.3)$$

$$+ \; \theta \; [CONV]^{n+1} + (1-\theta)[CONV]^n$$

$$+ \; \theta \; [DEC]^{n+1} + (1-\theta) \; [DEC]^n$$

where the diffusion, convection, and decay terms are, respectively :

$$DIFF \;=\; \oint_{cs} \left\{ \left(D_{xx}\frac{\partial C}{\partial x} + D_{xz}\frac{\partial C}{\partial z} \right) n_x + \left(D_{zx}\frac{\partial C}{\partial x} + D_{zz}\frac{\partial C}{\partial z} \right) n_z \right\} dl,$$

$$CONV \;=\; -\oint_{cs} C \left(u n_x + w n_z \right) dl,$$

$$DEC \;=\; -\lambda \, R_d \, C \, A_s,$$

and A_s is the area of the control volume, Δt is the computational time step, and θ is a weight parameter $(0 \le \theta \le 1)$ which is equal to 0.0 for explicit schemes and 1.0 for fully implicit schemes. If θ is equal to 0.5 we get the Crank-Nicolson scheme, which is second order time-accurate. Superscripts (n) and $(n+1)$ in $(0 \le \theta \le 1)$ equation (4.3) denote the evaluation of the terms at the time levels t (known values) and $t + \Delta t$ (unknown values), respectively.

The contour integration for the terms on the right hand side of the equation (4.3) is counter–clockwise. These terms are further discretized as described in the following sections.

4.3.1   Convection Term

The mid-point rule is used to approximate the convective term :

$$\oint_{cs} C(un_x + wn_z)dl \; = \; \sum_{k=1}^{4} C^{(k)} \left(u^{(k)}\Delta z^{(k)} - w^{(k)}\Delta x^{(k)} \right)$$

$$= \; \sum_{k=1}^{4} C^{(k)} F^{(k)} \tag{4.4}$$

where the superscript (k) refers to the edges of the control volume (see Chapter2, Figure 2.2), and $F^{(k)}$ is the fluid flux across edge k. The velocity components $u^{(k)}$ and $v^{(k)}$ are the averages of values at the end-points of edge (k).

The *outward* volume rate of flow through the edge k, $F^{(k)}$ is given by

$$F^{(k)} = (u^{(k)}\Delta z^{(k)} - w^{(k)}\Delta x^{(k)}) \tag{4.5}$$

To incorporate the second-order upwinding, $C^{(k)}$ in equation (4.4) is approximated at the mid-point of control-surface k by interpolation within the upwind control volume adjacent to the surface as discussed in Chapter 2. The interpolation scheme to obtain the $C^{(k)}$ value at the face k is based five-point shape functions (see Chapter 2) which are used for the iso-parametric interpolation

$$x \; = \; \sum_{i=1}^{5} N_i x_i \tag{4.6}$$

$$z \; = \; \sum_{i=1}^{5} N_i z_i \tag{4.7}$$

$$C \; = \; \sum_{i=1}^{5} N_i C_i \tag{4.8}$$

where the subscript i refers to the node number (see Chapter 2, Figure 2.2).

4.3.2   Diffusion Term

The diffusion term is discretized as given below

$$\oint_{cs} \left[(D_{xx}\frac{\partial C}{\partial x} + D_{xz}\frac{\partial C}{\partial z})n_x + (D_{zx}\frac{\partial C}{\partial x} + D_{zz}\frac{\partial C}{\partial z})n_z \right] dl$$

$$
\begin{aligned}
= \sum_{k=1}^{4} & \left[D_{xx}^{(k)} (\frac{\partial C}{\partial x})^{(k)} + D_{xz}^{(k)} (\frac{\partial C}{\partial z})^{(k)} \right] (\Delta z)^{(k)} \\
- \sum_{k=1}^{4} & \left[D_{zx}^{(k)} (\frac{\partial C}{\partial x})^{(k)} + D_{zz}^{(k)} (\frac{\partial C}{\partial z})^{(k)} \right] (\Delta x)^{(k)}
\end{aligned} \tag{4.9}
$$

where $D_{xx}^{(k)}$, $D_{zz}^{(k)}$ and $D_{xz}^{(k)}$ are again determined using averages of the end-point values and $(\frac{\partial C}{\partial x})^{(k)}$ and $(\frac{\partial C}{\partial z})^{(k)}$ are determined at the mid–point (in ξ, η, space) of the edge k using the derivatives of the shape functions, N_i.

$$
(\frac{\partial C}{\partial x})^{(k)} = \sum_{i=1}^{5} (\frac{\partial N_i}{\partial x})^{(k)} C_i \tag{4.10}
$$

$$
(\frac{\partial C}{\partial z})^{(k)} = \sum_{i=1}^{5} (\frac{\partial N_i}{\partial z})^{(k)} C_i \tag{4.11}
$$

where the derivatives of the shape-functions at the face centers are computed during the initialization procedure and stored for subsequent use.

4.3.3 Boundary Conditions

Fictitious cells are avoided in this application by using simple (i.e. first-order) upwinding as in chapter 3.

4.3.4 Solution Procedure

Finally, with the above formulation, the discretized equation for the solute transport equation (4.1) can be written as

$$
a_p C_p = \sum_{nb} a_{nb}\ C_{nb} + b \tag{4.12}
$$

where C_p is the (unknown) concentration value at the central node and C_{nb} the (unknown) values at the neighbours (including those for neighbouring control volumes introduced by upwinding) and b is the sum–total of known quantities. The coefficients a_p and a_{nb} are given in the Appendix-B. The Gauss-Seidel iteration technique is used to solve the discretized equation. To facilitate iterative convergence, the terms with

negative coefficients, are removed from the summation in (4.12), approximated by previous iteration values, and included in b. No difficulties were encountered in using this procedure.

4.4 Results

Three test cases of transient two-dimensional transport in porous media are considered in this section. In the first two cases, we compute the transport of a scalar in rectangular domains with three different types of boundary conditions. The solution with Crank-Nicolson and fully implicit time-stepping schemes are compared. In the third case, we consider non-rectangular domains, and nonorthogonal grids; we present methods for the suppression of spurious oscillations in the solutions, and demonstrate the schemes applicability for a wide range of Courant and grid Peclet numbers. In all cases, comparisons are done with known analytical solutions. We investigate the accuracy of different time-stepping schemes for OCV method and discuss the circumstances under which spurious oscillations arise, and methods for their removal.

4.4.1 Test Problem 1 : Dirichlet Boundary Condition at the Source

Test problem 1 considers unsteady, two-dimensional solute transport between two impervious boundaries. A finite-length strip solute source, whose concentration is a given function of time, is located asymmetrically along the z-axis at $x = 0$ in a unidirectional seepage velocity field as shown in Figure 4.1. The rectangular domain is 75 m in the x-direction and 50 m in the z-direction. The distances fixing the location of the source (see Figure 4.1) are : $2B = 10\ m$, $D_1 = 5\ m$ and $D_2 = 35\ m$. The uniform pore velocity, u is 0.1 m/day. The longitudinal, transverse, and cross dispersivities, D_{xx}, D_{zz} and D_{xz} are 1.0, 0.1, 0.0 m^2/day, respectively. The retardation factor, R_d is 1.0 and the decay coefficient, λ, is 0.0. The initial condition is given by $C(x, z, 0) = 0$, everywhere. The boundary condition at $x{=}0$, $t > 0$, is given by

$$C(0, z, t) = 0 \qquad 0 < z < D_1$$

$$C(0, z, t) = 1.0 \qquad D_1 < z < D_1 + 2B \qquad (4.13)$$

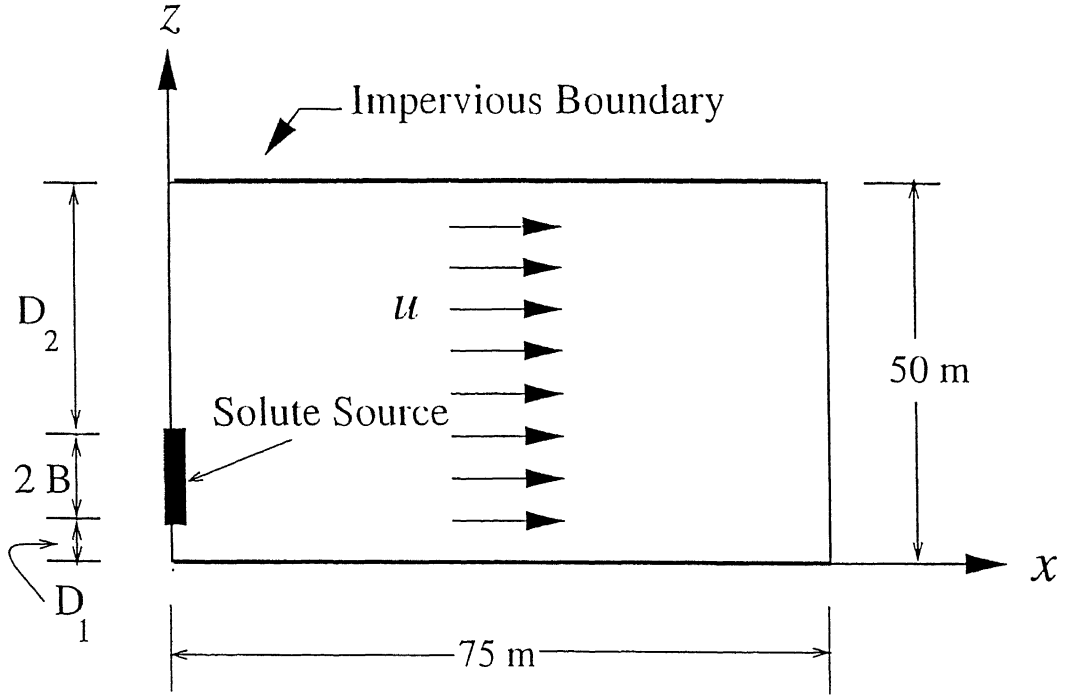$$C(0, z, t) = 0 \qquad D_1 + 2B < z < z_m.$$

Figure 4.1: Schematic of test problem 1

The analytical solution for the above problem is given by Batu[123]. The computational domain is represented by 61 × 41 (in the x, z directions respectively) grid points, and the computational time step, Δt, is one day (Courant Number $C_n = 0.08$, and grid Peclet Number $P_{e\Delta} = 0.125$). Here, and in the next test problem, we use the same grids and numerical parameters as were used by Batu[123,124] in the numerical validation of the analytical solutions. Figure 4.2 presents the longitudinal concentration distributions at $t = 100$ days as a function of x for $z = 10$ and 16.25 m. Figure 4.3 presents the lateral concentration distributions for $x = 5$ and 20 m. The numerical results in these figures are obtained using the Crank-Nicolson scheme ($\theta = 0.5$) and the fully Implicit scheme
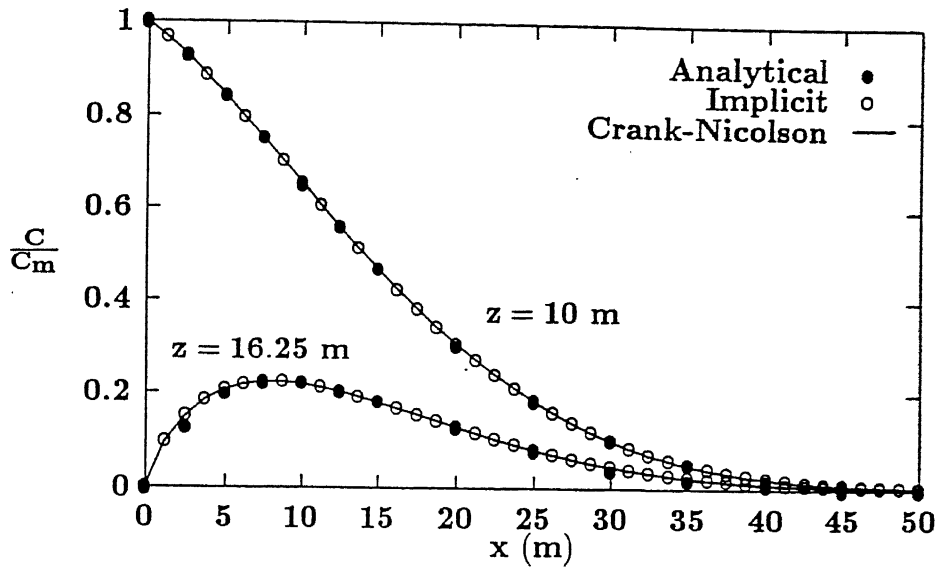
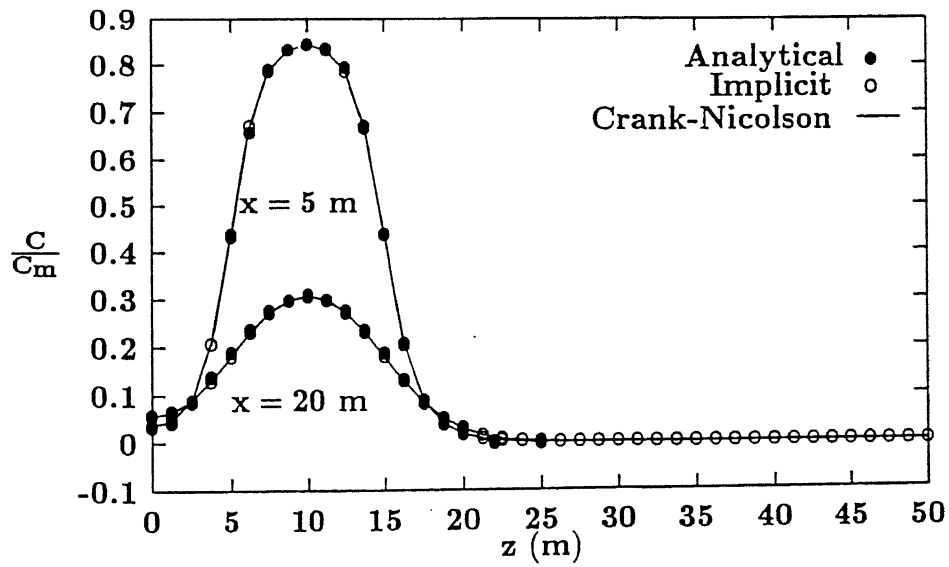Figure 4.2: Longitudinal concentration profile (Grid= 61 × 41 and Δt=1 day)

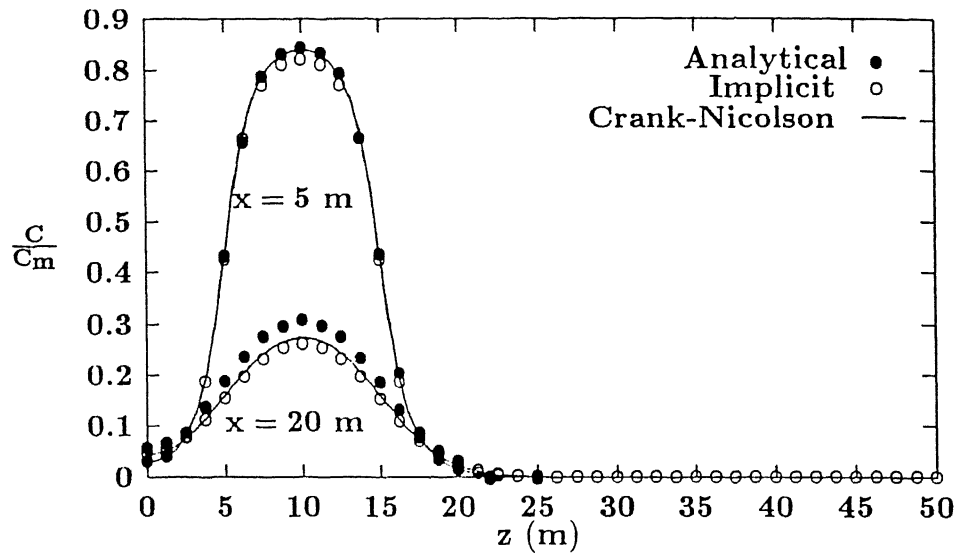Figure 4.3: Transverse concentration profile (Grid= 61 × 41 and Δt=1 day)

Figure 4.4: Transverse concentration profile (Grid= 61 × 41 and Δt=10 day)

(θ=1.0). As can be seen from these figures, the method gives accurate solutions. To examine numerical stability for Courant numbers greater than unity, computations are made with 121 × 81 grid points and Δt = 10 days (C_n = 1.6). The computed results for this case obtained using the implicit scheme are compared with the analytical results in Figure 4.4. Although stable results could be obtained using the OCV scheme, there is somewhat more error in the numerical solution. This is expected since the most accurate results are generally obtained when C_n =1. The fully Implicit scheme, which is only first order accurate in time, introduces noticeably larger errors than the Crank-Nicolson scheme.

4.4.2 Test Problem 2 : Mixed Boundary Condition at the Source

This example considers the two-dimensional solute transport in a unidirectional flow field with mixed (Robin) type boundary conditions at the source. A finite-length strip solute source is located asymmetrically along the z-axis at x=0 in a unidirectional velocity field as shown in Figure 4.5. The domain is 185 m in the x-direction and

$(z_m =)53\ m$ in the z-direction. The other dimensions (see Figure 4.5) are : $2B = 5$ m, $D_1 = 45\ m$, $D_2 = 3\ m$. The uniform pore velocity, u is equal to $0.15\ m/day$. The longitudinal, transverse, and cross dispersion coefficients, D_{xx}, D_{zz}, D_{xz} are equal to $3.195, 0.645, 0.0\ m^2/day$, respectively. The retardation factor and the decay coefficient are equal to 1.0 and 0.0, respectively.
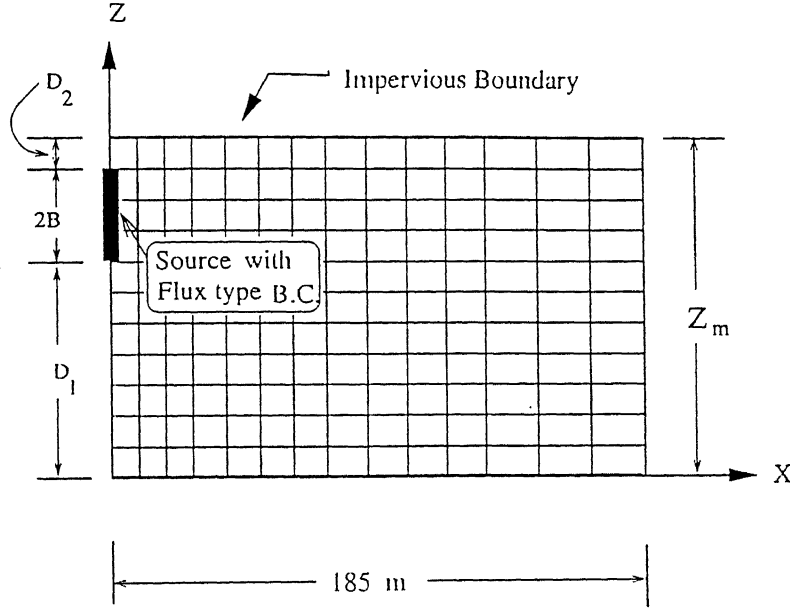
Figure 4.5: Schematic of test problem 2

The initial condition is given by $C(x, z, 0) = 0$. Boundary condition is given by

$$
\begin{array}{lll}
F_x(0, z, t) & = 0 & 0 < z < D_1 \\
F_x(0, z, t) & = uC_m & D_1 < z < D_1 + 2B \\
F_x(0, z, t) & = 0 & D_1 + 2B < z < z_m
\end{array}
\qquad (4.14)
$$

where, $C_m = 1.0$ and F_x is the convective-dispersive flux component (the mass flow rate of solute per unit area) in the x-direction :

$$
F_x = \Phi uC - \Phi D_{xx}\frac{\partial C}{\partial x}
$$

$$
F_z = -\Phi D_{zz}\frac{\partial C}{\partial z}
$$

where, Φ is the porosity ($=0.25$ for the problem under consideration).

The analytical solution for the above problem is given by Batu[124]. In the numerical solution, the computational domain is divided into 48 × 29 non-uniform grid points (closer mesh spacing near the upstream boundary) and Δt is equal to one day (C_n = 0.04, $P_{e\Delta}$ = 0.18). The numerical results for the normalized concentration obtained using the Crank-Nicolson scheme (θ = 0.5) are compared with the analytical solution in Figures 4.6, 4.7, and 4.8. Longitudinal and transverse concentration versus distance comparison at z= 51.5 m and x= 22.5 m are shown in Figures 4.6 and 4.7, respectively, at a 180 day period. Figure 4.8 presents the time variation of the normalized concentration at the point (x, z)= (8.75, 47.5m). As can be seen from these figures, a good correspondence exists between the numerical results and the analytical results.
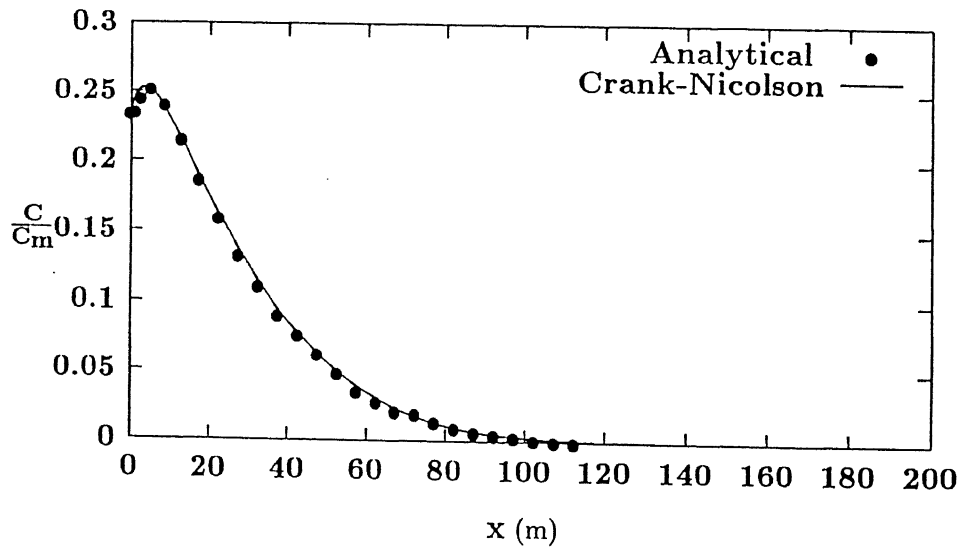

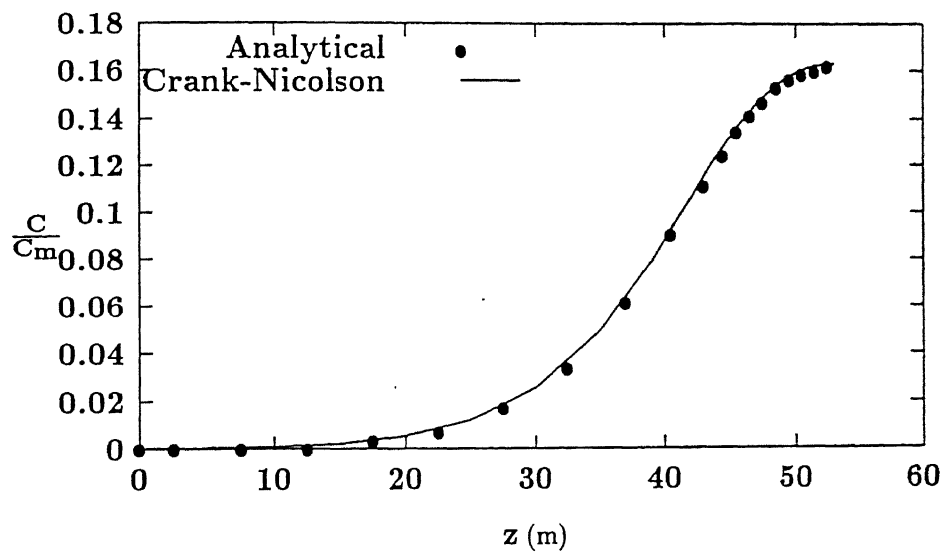Figure 4.6: Longitudinal concentration profile at z=51.5 m

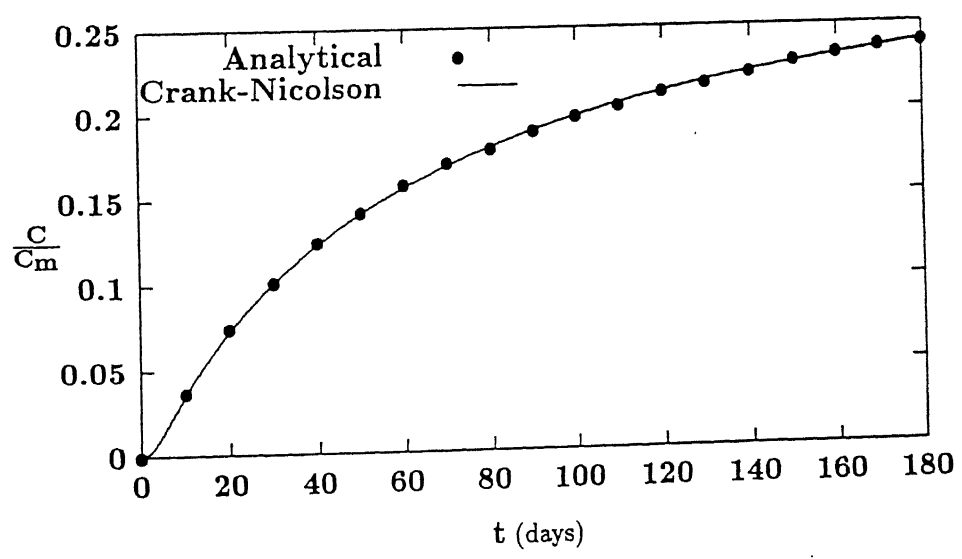Figure 4.7: Longitudinal concentration profile at x=22.5 m

Figure 4.8: Temporal variation at x=8.75, z=47.5

4.4.3   Test Problem 3 : Non-rectangular Domain

In this example, a non-rectangular domain with a non-orthogonal grid as shown in
Figure 4.9 is considered. The length of the domain along the x-axis, L, is 10 m, its
lateral boundaries are given by the equation

$$z = \pm \left[2.0 + 1.05sin(\frac{\pi x}{2L})\right] \tag{4.15}$$

A strip source of length 1 m is placed center-symmetrically along the z-axis at $x = 0$ in
a uniform velocity field. The retardation factor and the decay coefficients are equal to
1.0 and 0.0, respectively and $u= 0.2\ m/day$. The initial condition is given by $C(x, z, 0)$
$= 0$, everywhere. The boundary condition at $x = 0$ is given by

$$
\begin{aligned}
C(0, z, t) &= 1 & \text{at the source} \\
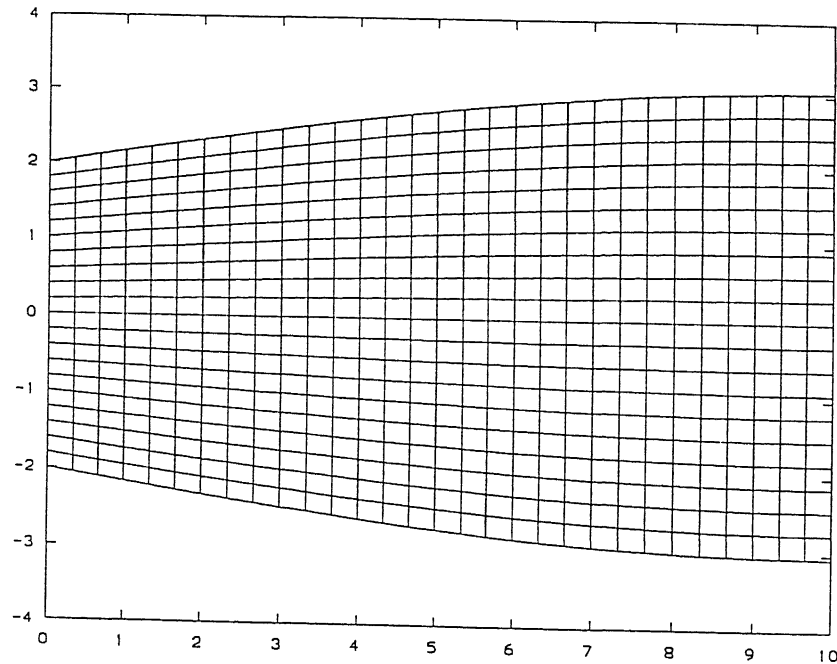C(0, z, t) &= 0 & \text{else where}
\end{aligned}
\tag{4.16}
$$

Figure 4.9: Non-orthogonal grid for test problem 3

Dirichlet boundary conditions are applied at the lateral boundaries. For the specifi-
cation of the time varying concentration along the lateral boundaries, the analytical
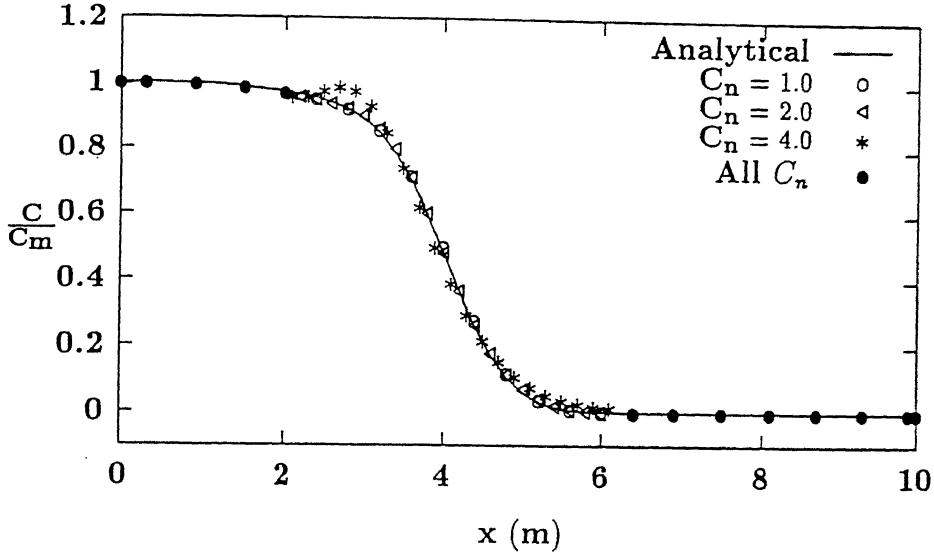
Figure 4.10: Results for Crank-Nicolson scheme $(Pe_\Delta=2)$

solution given by Javendel et al.[125] for the semi-infinite domain is used. The analytical solution can be written as

$$C(x,z,t) = \frac{x}{4\sqrt{(\pi D_{xx})}}exp(\frac{ux}{2D_{xx}})\int_0^t exp\left[-\frac{u^2t}{4D_{xx}}-\frac{x^2}{4D_{zz}t}\right]\frac{1}{t^{3/2}}$$

$$\left[erf(\frac{a-z}{2\sqrt{(D_{zz}t)}})+erf(\frac{a+z}{2\sqrt{(D_{zz}t)}})\right]dt$$

where, a is the half width of the strip source at $x=0$. A fourth-order numerical integration is used to evaluate the above equation to obtain the concentration at the lateral boundaries, to supply the boundary conditions, as well as at any interior point if needed for comparison with the numerical solution. The computational domain is divided into 101×81 grid points. The time step Δt is taken as 0.1, 0.5, 1.0 and 2.0 days and the corresponding Courant number, C_n $(\equiv \frac{u\Delta t}{\Delta x})$ is 0.2, 1.0, 2.0 and 4.0, respectively.

In the previous sections Crank-Nicolson scheme has been shown to be more accurate than fully Implicit time-stepping. Therefore, we first investigate the solutions obtained using Crank-Nicolson scheme for this problem. Figure 4.10 shows the Crank-Nicolson
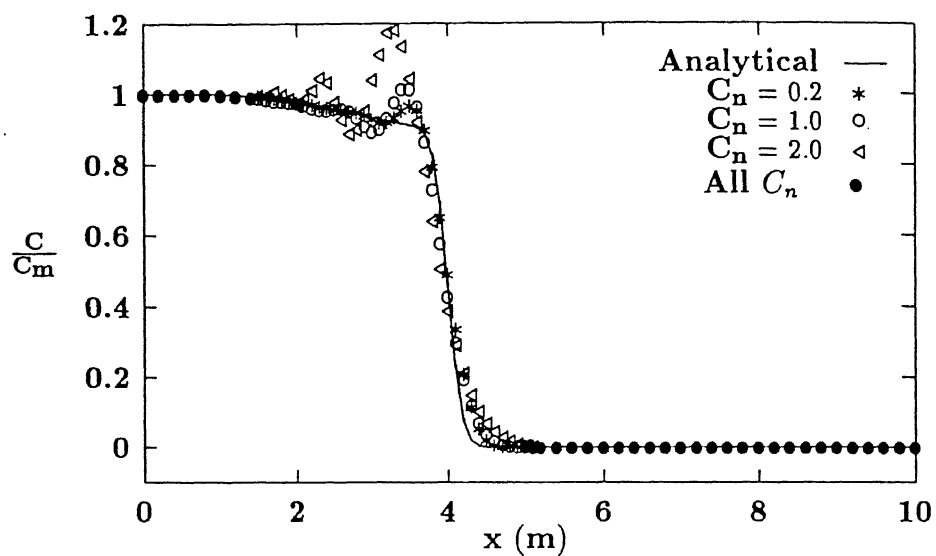
Figure 4.11: Results for Crank-Nicolson scheme $(Pe_\Delta=40)$
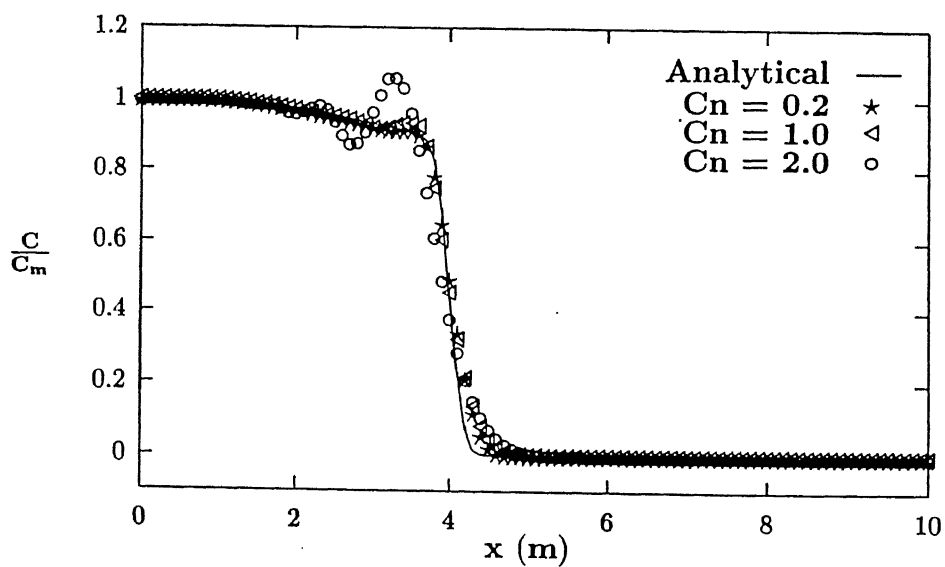
Figure 4.12: Results for Crank-Nicolson scheme with flux limiter $(Pe_\Delta=40)$
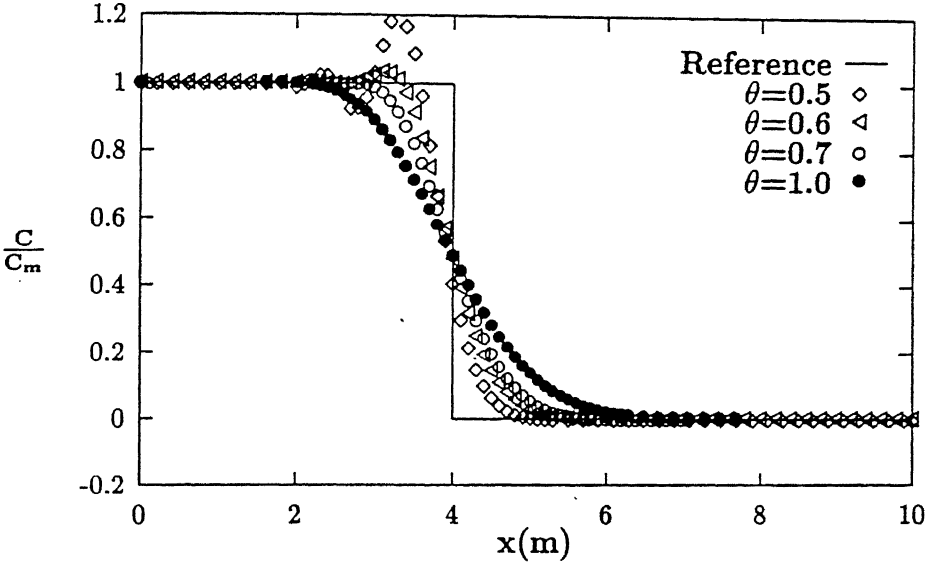
Figure 4.13: Results on grid corresponding to figure 10 ($C_n{=}2$, $Pe_\Delta{=}2000$)
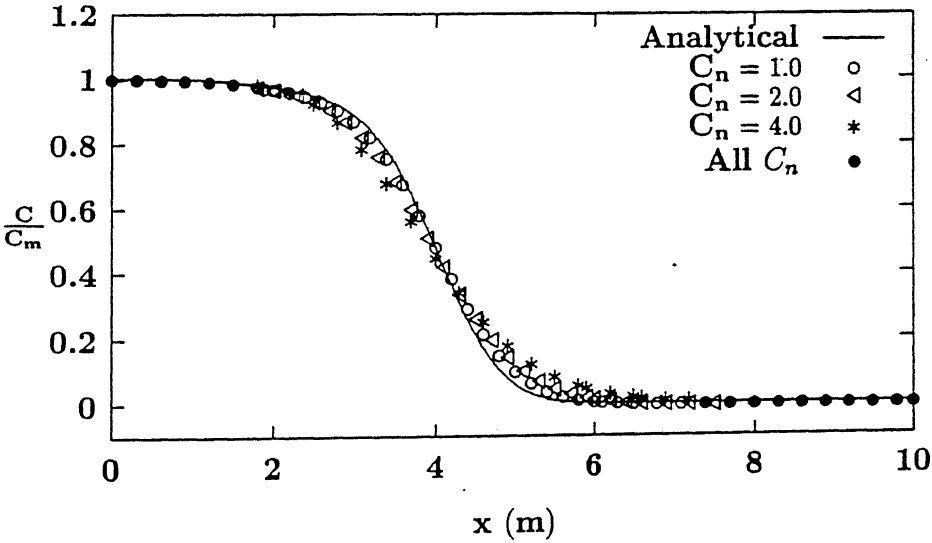
Figure 4.14: Results for a scheme corresponding to $\theta{=}0.7$ ($Pe_\Delta{=}2$)
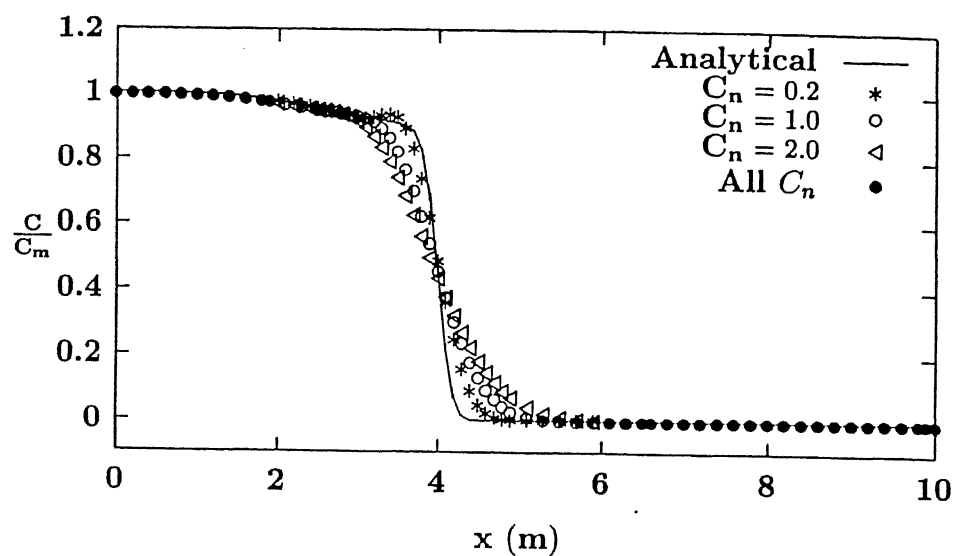
Figure 4.15: Results for a scheme corresponding to $\theta=0.7$ $(Pe_\Delta = 40)$
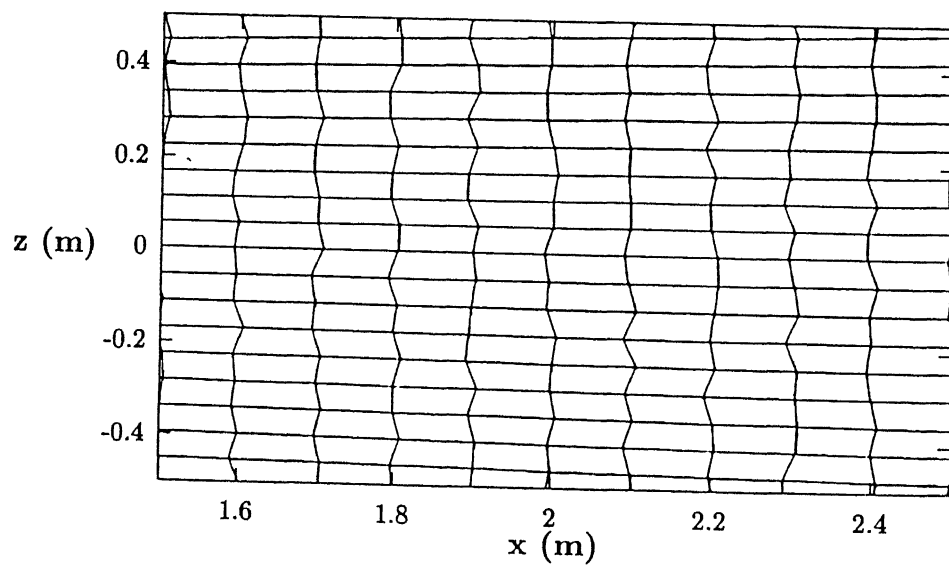
Figure 4.16: Distorted grid for test problem 3
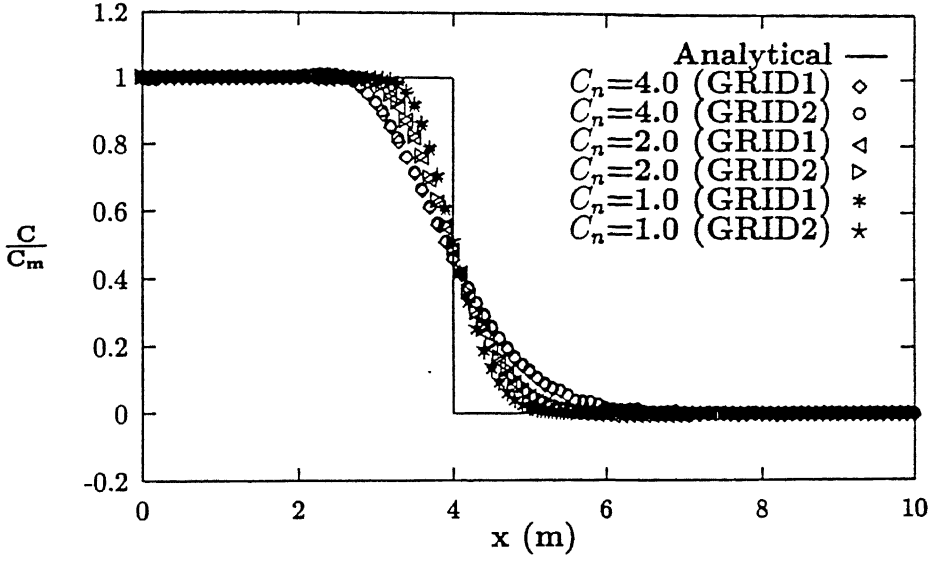
Figure 4.17: Results for a scheme with $\theta=0.7$ and $Pe_\Delta=2000$ on GRID1 (fig. 4.9) and GRID2 (fig. 4.16)
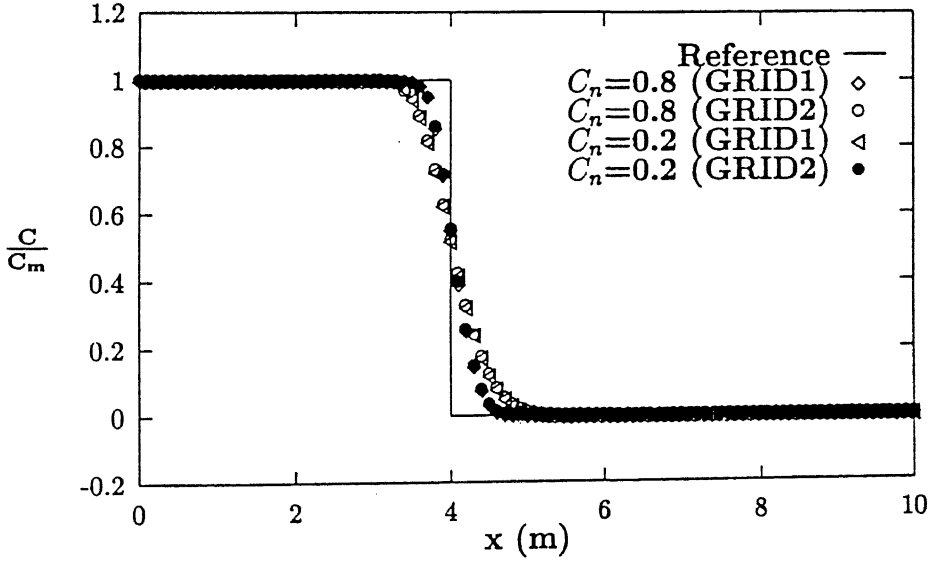
Figure 4.18: Results for a scheme with $\theta=0.7$ and $Pe_\Delta=2000$ on GRID1 (fig. 4.9) and GRID2 (fig. 4.16)

solution, for various Courant numbers, for the problem at $t=$ 20 days; the longitudinal grid Peclet number ($Pe_\Delta = \frac{u\Delta x}{D_{xx}}$) is a moderate value of 2.0 (D_{xx}=0.01 m^2/day, D_{zz}=0.0025 m^2/day). The solutions for Courant numbers of 1.0 and 2.0 are good, while the solutions for $C_n = 4.0$ is acceptable, showing only a slight overshoot. The solutions for C_n below unity all fall on the analytical solution, but are not shown (to avoid clutter).

Figure 4.11 shows for the same situations, the Crank-Nicolson solutions for a grid Peclet number of 40.0 (D_{xx}=0.0005 m^2/day, D_{zz}=0.0025 m^2/day)— corresponding to a highly convection dominated flow. It can be seen that for this case, the solutions show overshoots – small for Courant numbers below unity but large for higher Courant numbers. This latter errors may be unacceptable in many cases.

Next, in an attempt to suppress spurious oscillations, we use a flux limiter described in chapter 3. Figure 4.12 shows numerical results for the same case ($Pe_\Delta = 40$) as previously studied but with the flux limiter in use. The results clearly shows that for Courant numbers of unity, and below, the flux limiter is successful in removing oscillations. But for higher Courant number flux limiter fails. This itself is not surprising as the above flux limiter, like most flux limiters, uses only the neighbouring cell values to detect oscillations, while at high Courant numbers information arrives from far cells. In general, flux limiters have been used for explicit schemes with Courant numbers below unity. No flux limiters used with Implicit schemes have yet been demonstrated as effective for high Courant numbers.

A different strategy can be used to avoid overshoots by introducing enough numerical dissipation into the scheme by choosing the time-stepping parameter θ as some value other than 0.5. As θ values below 0.5 do not allow unconditionally stable schemes, we will use only values above 0.5. To use values other than θ=0.5 is to introduce a first-order error, and numerical diffusion, into the time-stepping, therefore we would like to make the least departure from that value as possible.

Figure 4.13 shows the solutions, for Courant number C_n =2.0, with θ chosen variously, for a very high grid Peclet number of 2000.0 ($D_{xx} = 0.00001, D_{zz} = 0.0$). The figure shows θ=0.5, Crank-Nicolson, gives substantial overshoots which decrease, but do not

disappear, for θ =0.6. The value of θ=0.7 does not show any overshoots. To increase θ further would only increase the numerical spreading, as can be seen by the θ=1.0 solution in the figure. Therefore, for this case atleast, θ=0.7 seems an optimum compromise between numerical dispersion (oscillations) and numerical dissipation (spreading). We now show that the scheme performs satisfactorily, with θ=0.7, for a wide range of Courant and grid Peclet numbers.

Figure 4.14 and 4.15 show the same solutions as figures 4.10 and 4.11, with Pe_Δ (grid Peclet number)= 2 and 40, respectively, but with solutions time-stepped with θ=0.7 instead of the Crank-Nicolson ($\theta = 0.5$) scheme. The solutions show no overshoots but have slightly more spreading than the Crank-Nicolson solutions.

The OCV scheme has been shown to lose little of its accuracy on mildly non-orthogonal grids, when applied to steady-state problems (see chapter 2 and 3). Here we show that this feature is retained by the present implicitly time-stepped scheme : A new distorted grid is generated by perturbing each interior grid points in the figure 10 randomly between $\pm$ 10 percent of the grid interval, in x-direction. A section of the distorted grid is shown in figure 4.16. Figure 4.17 shows the solution at t= 20 days, on the grids shown in to figure 4.9 (GRID1) and 4.16 (GRID2) for Pe_Δ=2000 and C_n= 1.0, 2.0 and 4.0. There is little deterioration of accuracy due to grid distortion. This holds true even for C_n below unity as is shown in figure 4.18. The figures also show that numerical spreading of the solution increases unambiguously with Courant number, as can be expected. The results in figures 4.17 and 4.18 are obtained using the scheme with θ=0.7 *and* the flux limiter. The flux limiter does not seem to affect the final solution for Courant number greater than unity but *does* stabilize the time-stepping scheme. For example, in this case without the flux-limiter we could not get convergence beyond C_n= 2. (However, *if* convergence is obtained for $C_n > 1$, the solutions with and without flux limiter are almost the same).

4.5   Summary

In this study, an overlapping control volume (OCV) technique is used for solving the two-dimensional, transient, solute-transport equation for ground water flows. Two-

dimensional domains with orthogonal and non-orthogonal grids are considered. The salient features of this chapter are summarized below

- An isoparametric formulation is used to compute diffusion and to introduce higher-order upwinding.

- An implicit formulation is used for time stepping.

- The numerical technique is validated using the two-dimensional analytical solutions. The test cases include Dirichlet, Neumann and mixed boundary conditions.

- It is shown that Crank-Nicolson scheme is most accurate for diffusion dominated flows (with grid Peclet number less than 2), but introduces spurious oscillations for convection dominated flows.

- The flux limiter used in this study can remove the oscillations for Courant numbers below unity. For higher Courant numbers, spurious oscillations can be avoided by using an appropriate value (θ=0.7) for the implicit weighing factor.

- The resulting scheme is shown to work well for a wide range of grid Peclet numbers and Courant numbers well in excess of 1.

- The effect on accuracy of mild non-orthogonality of the grids is shown to be insignificant.

Chapter 5

A Solution Method of Navier-Stokes Equation on Nonstaggered Grids

---

5.1   Introduction

In this chapter, an OCV algorithm for computing steady and unsteady solutions to the two-dimensional incompressible Navier-Stokes equations on nonstaggered grids is presented. An isoparametric interpolation is used to compute diffusion terms and to introduce higher-order upwinding. Since the problem is solved in the physical domain, the governing equations are much simpler than in the generalized coordinate system. An equal-order interpolation is used for the velocity components and pressure. The problem of velocity-pressure decoupling is avoided by using momentum interpolation[12]. Cartesian components of velocities are used in the calculations. This algorithm can be used to solve problems on nonorthogonal structured grids.

5.2   Formulation

The solution domain is discretized into a structured non-orthogonal grid as shown in Figure 2.1 (see Chapter 2). A typical control volume is shown by the shaded area in the figure and also in Figure 2.2. We refer to these control volumes by the index of this

central node, e.g., the control volume for (i, j) is shown in Figure 2.2. A non-staggered arrangement is used for the dependent variables, i.e. the pressure and each of the velocity components are defined at the same grid points.

5.2.1 Governing Equations

The two-dimensional Navier-Stokes equations for laminar incompressible flow for an arbitrary control volume bounded by a closed line l can be expressed in the following integral form:

$$\oint_{cs} (\rho u_q) dl_q = 0 \tag{5.1}$$

$$\frac{\partial}{\partial t} \int \int (\rho u_p) dA + \oint_{cs} (\rho u_q u_p) dl_q \; - \; \oint \mu \left(\frac{\partial u_p}{\partial x_q} + \frac{\partial u_q}{\partial x_p} \right) dl_q$$

$$+ \; \oint p \, dl_p - \int \int S_{u_p} dA = 0 \tag{5.2}$$

where the Einstein summation convention is used and $p = 1, 2$ corresponds to the u_1, u_2 momentum equations for the x_1, x_2 directions, respectively. The components of the outward *normal* line element are dl_q and counterclockwise contour integration is the assumed. Here ρ represents the density of the fluid and S_{u_p} is the source term.

5.2.2 Discretization Procedure

The discretization procedure for the continuity and momentum equations are described below.

Continuity Equation

Equation (5.1) is discretized in the following way.

$$\oint_{cs} (\rho u_q) dl_q \approx \sum_{k=1,2,3,4} (\rho u_1 dl_1 + \rho u_2 dl_2)^{(k)} = \sum_k (\rho u_1 \Delta x_2 - \rho u_2 \Delta x_1)^{(k)} \tag{5.3}$$

where the superscript (k) refers to the edges of the control volume [shown circled in Figure 2.2], $(\Delta x_2 \mathbf{i} - \Delta x_1 \mathbf{j})^{(k)}$ is the normal surface vector of the kth cell edge, with

$\Delta x_1^{(k)}$ and $\Delta x_2^{(k)}$ being the change in coordinates along edge k in the counterclockwise direction, and $u_1^{(k)}$ and $u_2^{(k)}$ are the velocity components defined at the mid-point of edge k.

In discretized form, the continuity equation can be expressed as

$$\sum_k F^{(k)} = F^{(1)} + F^{(2)} + F^{(3)} + F^{(4)} = 0, \tag{5.4}$$

where the $F^{(k)}$ is the outward mass flux through face (k):

$$F^{(k)} \equiv \rho u_1^{(k)} \Delta x_2^{(k)} - \rho u_2^{(k)} \Delta x_1^{(k)}. \tag{5.5}$$

Rate of change term

The lumped mass approach is used for the transient term. The value of the velocity component u_1 at the central node P of the control volume (CV) represents an average over the CV as a whole. Thus

$$\frac{\partial}{\partial t} \int \int \rho u_p dA \approx \frac{(\rho u_p A_s)^{n+1} - (\rho u_p A_s)^n}{\Delta t} = \rho A_s \frac{(u_p)^{n+1} - (u_p)^n}{\Delta t} \tag{5.6}$$

where A_s is the area of the cell.

Convection Terms

The surface integral over the convection terms for u_p momentum equation is approximated using the midpoint rule; we get

$$\oint_{cs} (\rho u_q u_p) dl_q \approx \sum_k (\rho u_p u_1 \Delta x_2 - \rho u_p u_2 \Delta x_1)^{(k)}$$

$$= \sum_k u_p^{(k)} F^{(k)} \equiv \sum_k F_{pc}^{(k)} \tag{5.7}$$

where $u_p^{(k)}$ is the value of u_p at the center of edge (k), and $F_{pc}^{(k)}$ is the outward *convected flux* of components u_p through edge k. To incorporate upwinding, $u_p^{(k)}$ in (5.7) is approximated at the mid-point of control-surface k by interpolation within the upwind control volume at that surface. This scheme for convective modelling is conservative.

The method used for interpolation is based on finite–element type shape functions as explained in chapter 2. The iso-parametric formulation is used :

$$u_1 = \sum_{i=1}^{5} (u_1)_i N_i \qquad (5.8)$$

$$u_2 = \sum_{i=1}^{5} (u_2)_i N_i \qquad (5.9)$$

$$x_1 = \sum_{i=1}^{5} (x_1)_i N_i \qquad (5.10)$$

$$x_2 = \sum_{i=1}^{5} (x_2)_i N_i \qquad (5.11)$$

where the $(x_1)_i$'s and $(x_2)_i$'s are the coordinates of the 5 grid points making the control volume. The equations (5.10) and (5.11) map the control volume in Figure 2.2 onto the transformed control volume in (ξ, η) coordinates shown in Figure 2.3 and the N_i's are shape functions defined in terms of these coordinates (see Chapter 2). For example

$$N_1(\xi, \eta) = \frac{1}{4}(\xi\eta - \xi - \eta) + \frac{1}{8}(\xi^2 + \eta^2),$$

etc.

For the purposes of upwinding, $u_p^{(k)}$ is found by using (5.8) to interpolate the value at the mid–point of the edge k in the transformed control volume.

To compute partial derivatives $\frac{\partial u_1}{\partial x_1}$, $\frac{\partial u_1}{\partial x_2}$, etc, we need to compute $\frac{\partial N_i}{\partial x_1}$, $\frac{\partial N_i}{\partial x_2}$ at the midpoint of the edges of the control volumes. These latter values are computed as part of the initialization procedure and stored for subsequent computations. The derivatives of any variable (ϕ) are determined from :

$$\frac{\partial \phi}{\partial x_1} = \sum_{i=1}^{5} \phi_i \frac{\partial N_i}{\partial x_1} \qquad (5.12)$$

$$\frac{\partial \phi}{\partial x_2} = \sum_{i=1}^{5} \phi_i \frac{\partial N_i}{\partial x_2}. \qquad (5.13)$$

where the ϕ_i's are the values of ϕ at the ith node.

Diffusion flux

This term is also approximated using the mid-point rule. The discretized term from the u_1 momentum equation (5.2) is represented as

$$\oint \mu \left(\frac{\partial u_1}{\partial x_q} + \frac{\partial u_q}{\partial x_1} \right) dl_q \approx \sum_k \left(2\mu \frac{\partial u_1}{\partial x_1} \Delta x_2 - \mu \left(\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1} \right) \Delta x_1 \right)^{(k)}$$

$$= \sum_k \left[2\mu \sum_j (u_1)_j \frac{\partial N_j}{\partial x_1} \Delta x_2 - \sum_j \mu \left((u_1)_j \frac{\partial N_j}{\partial x_2} + (u_2)_j \frac{\partial N_j}{\partial x_1} \right) \Delta x_1 \right]^{(k)}$$

$$\equiv - \sum_k (F_{1d})^{(k)} \tag{5.14}$$

where $F_{1d}^{(k)}$ is the diffusion flux of u_1 through edge k and is represented by the quantity in the square brackets. The diffusion flux of u_2, $F_{2d}^{(k)}$, is similarly represented in terms of (u_1, u_2) values at the nodes. The derivatives of the shape functions are evaluated at the mid-points, in (ξ, η) space, of the control surfaces. The summation is carried out over all the surfaces of the control volume. The diffusion modelling in OCV is second-order accurate and has exactly the same error as the conventional five-point center-difference discretization on a uniform grid.

Source Term

It is assumed that the value at the central node represents the mean value over the whole control volume. Thus, we can write the source term as

$$\int \int S_{u_p} dA \approx (S_{u_p})_{ij} A_s$$

Apart from the real source S_{u_p}, explicitly treated parts of the convection and diffusion fluxes may also be added to S_{u_p}, during iterations for the steady-state or implicit solutions.

Pressure Term

This term is treated explicitly in the predictor step (see below). For the u_1-momentum equation the pressure term is

$$-\oint_{cs} p \, dl_1 \approx -\sum_k p^{(k)} \Delta x_2^{(k)},$$

and for the u_2 equation it is

$$-\oint_{cs} p \, dl_2 \approx \sum_k p^{(k)} \Delta x_2^{(k)},$$

where $p^{(k)}$ is the pressure at the kth face center. In the correction step, gradient terms of the pressure corrections arise which are treated analogously to the diffusion flux. This is explained in a later section (5.3.2).

5.3   Time Integration Scheme

Euler semi-explicit or implicit time integration schemes are usually used to discretize the time dependent incompressible Navier-Stokes equations. In semi-explicit schemes, the momentum equations are discretized in an explicit manner, except for the pressure gradient terms which are treated implicitly; the continuity equation is also enforced implicitly. As a consequence, the pressure-velocity coupling reduces to a Poisson equation for the pressure corrections. Such schemes, because of their reliance on explicit differences, suffer from time-step restrictions. In the implicit schemes, the equations are discretized fully implicitly, with the coupling being determined by the momentum equations. Some methods which fall within this latter group employ sequential iteration (such as the SIMPLE method), in which the equations for each variable are repeatedly solved in succession.

5.3.1   Semi-Explicit Time-Stepping

We adopt a semi-explicit scheme in which the discretized equations

$$\rho(A_s)\frac{u_1^{n+1} - u_1^n}{\Delta t} + \sum_k (F_{1c}^n + F_{1d}^n)^{(k)} = -\sum_k [p^{n+1}]^{(k)} \Delta x_2^{(k)} + S_{u_1}^n A_s \qquad (5.15)$$

$$\rho(A_s)\frac{u_2^{n+1} - u_2^n}{\Delta t} + \sum_k (F_{2c}^n + F_{2d}^n)^{(k)} = \sum_k [p^{n+1}]^{(k)} \Delta x_1^{(k)} + S_{u_2}^n A_s \qquad (5.16)$$

have to be solved along with

$$\sum_k [F^{n+1}]^{(k)} = 0, \qquad (5.17)$$

for each finite volume cell, where the superscript n refers to the (known) values at the current-time step and $n+1$ to the (unknown) values at the next time level.

We adopt a two step process. First, a predicted velocity u_1^* and u_2^* are found using the known n level values :

$$\rho(A_s)\frac{u_1^* - u_1^n}{\Delta t} + \sum_k (F_{1c}^n + F_{1d}^n)^{(k)} = -\sum_k [p^n]^{(k)} \Delta x_2^{(k)} + S_{u_1}^n A_s \qquad (5.18)$$

$$\rho(A_s)\frac{u_2^* - u_2^n}{\Delta t} + \sum_k (F_{2c}^n + F_{2d}^n)^{(k)} = \sum_k [p^n]^{(k)} \Delta x_1^{(k)} + S_{u_2}^n A_s. \qquad (5.19)$$

Subtracting equation (5.18) from (5.15), and (5.19) from (5.16), we get equations

$$\rho(A_s)\frac{u_1'}{\Delta t} = -\sum_k (p')^{(k)} \Delta x_2^{(k)}, \qquad (5.20)$$

$$and \quad \rho(A_s)\frac{u_2'}{\Delta t} = \sum_k (p')^{(k)} \Delta x_1^{(k)}, \qquad (5.21)$$

for the corrections

$$u_1' = u_1^{n+1} - u_1^n, \qquad u_2' = u_2^{n+1} - u_2^n, \qquad p' = p^{n+1} - p^n. \qquad (5.22)$$

The corresponding flux corrections $(F')^{(k)}$ have to satisfy

$$\sum_k (F')^{(k)} = -\sum_k (F^*)^{(k)} \qquad (5.23)$$

where $(F^*)^{(k)}$ is the massflux corresponding to the predicted velocities and $(F')^{(k)}$ is the perturbation due to the corrections in velocity and pressure. Equations (5.20), (5.21), and (5.23), have to to be simultaneously solved by iterations. This cycle of the iterations constitute the *corrector* step in the time-stepping. When the iterations converge the corrected velocity values are obtained from

$$u_1^{n+1} = u_1^n + u_1'$$
$$u_2^{n+1} = u_2^n + u_2',$$

and the calculations proceed to the next time-step.

5.3.2   Pressure-Velocity Corrections

Since a nonstaggered, or collocated, arrangement is used in this formulation, the pressure-velocity decoupling or checkerboard pressure distribution may occur if the variables (velocities and pressure) at the cell edges are calculated by linear interpolation[4]. Rhie and Chow[12] used *momentum interpolation* to overcome this problem and opened a way towards the use of collocated grids for the solution of Navier-Stokes equation for incompressible flows in complex geometries. We, following similar lines, use a formulation in which the velocity at the cell faces are computed by allowing linear interpolation of the convective and diffusive terms but not of the pressure term. This is done in the following way:

If the discretized equation for predicted velocity in x_1 direction is written as

$$u_1^* = u_1^n + \frac{\Delta t}{\rho(A_s)} \left(-\sum_k (F_{1c} + F_{1d})^{(k)} + S_{u_1} - \sum_k p^{(k)} \Delta x_2^{(k)} \right)^n \qquad (5.24)$$

where all terms in the right are n^{th} time-step values. We define a new variable U_1 which is computed using

$$U_1^* = u_1^n + \frac{\Delta t}{\rho(A_s)} \left(-\sum_k (F_{1c} + F_{1d})^{(k)} + S_{u_1} \right)^n , \qquad (5.25)$$

i.e., without the pressure term present in (5.24).

Now, to estimate the mass flux $(F^*)^{(1)}$ at the edge 1 (see Figure 2.3), straight forward linear interpolation would use

$$(F^*)_{lin}^{(1)} = \rho \overline{[((u_1)_{i-1,j}, (u_1)_{i,j-1})]} \Delta x_2^{(1)} - \rho \overline{[((u_2)_{i-1,j}, (u_2)_{i,j-1})]} \Delta x_1^{(1)} \qquad (5.26)$$

where the over-bar indicates a linear interpolation. But we use

$$\begin{aligned}
(F^*)^{(1)} &= \rho \overline{[((U_1)_{i-1,j}, (U_1)_{i,j-1})]} \Delta x_2^{(1)} - \rho \overline{[((U_2)_{i-1,j}, (U_2)_{i,j-1})]} \Delta x_1^{(1)} \\
&\quad -\Delta t \left[\frac{\partial p}{\partial x_1} \right]^{(1)} \Delta x_2^{(1)} + \Delta t \left[\frac{\partial p}{\partial x_2} \right]^{(1)} \Delta x_1^{(1)}
\end{aligned} \qquad (5.27)$$

where the pressure derivatives are values at the mid-point of edge 1, and are estimated using equations (5.12) and (5.13). This method thus avoids the linear interpolation

of pressure implicit in (5.26), rather it separates the pressure term using (5.25), and later incorporates it as a gradient term at the face, in equation (5.27). This seemingly simple alteration effectively avoids pressure-velocity decoupling.

5.3.3 Time-Stepping Algorithm

1. Initial conditions for velocity and pressure are prescribed; shape functions and derivative values are computed for each cell, etc..

2. Equations (5.24) and (5.25) are used to compute the *predicted* cell-center u_p and U_p for the next time step.

3. The *predicted* mass flux through each cell edge k is computed using

$$(F^*)^{(k)} = \sum_j \rho U_j^{(k)} l_j^{(k)} - \sum_j \Delta t \left(\frac{\partial p}{\partial x_j}\right)^{(k)} l_j^{(k)}, \qquad (5.28)$$

where $U_j^{(k)}$ is the linearly interpolated face value of U.

4. The flux correction at the edge k are computed by

$$(F')^{(k)} = \rho \Delta x_2(k) u_1'^{(k)} - \rho \Delta x_1^{(k)} u_2'^{(k)} - \Delta t \sum_j \left(\frac{\partial p'}{\partial x_j}\right)^{(k)} l_j^{(k)}, \qquad (5.29)$$

using linear interpolation to obtain the face-center quantities $u_1^{(k)}$ and $u_2^{(k)}$ in (5.29)and the iso-parametric form (equation 5.12-5.13) to obtain the gradient of p' at the face (k). (Initially $p'=0$, for each time-step).

5. The mass flux residual for each cell is computed by

$$\Re = -\sum_k (F^*)^{(k)} - \sum_k (F')^{(k)} \qquad (5.30)$$

6. The pressure correction for each node (i, j) is obtained from the relation

$$p' \leftarrow p' + \omega \frac{\Re}{a_{i,j}} \qquad (5.31)$$

where ω is relaxation factor and $a_{i,j}$ is the diagonal coefficient obtained from the discretization of pressure in equations (5.28)-(5.30).

7. If $\Re_{rms} > \epsilon$ goto step 4

8. Store the updated mass flux through cell-faces from

$$(F^*)^{(k)} \leftarrow (F^*)^{(k)} - \Delta t \sum_j \left(\frac{\partial p'}{\partial x_j} \right)^{(k)} l_j^{(k)} , \qquad (5.32)$$

(to be used to compute predicted velocities for the next time-step).

9. Update pressure at the nodes (i, j) node $p \leftarrow p + p'$

10. Update velocity at nodes

$$u_j' = -\frac{\Delta t}{\rho(A_s)} \sum_k (p')^{(k)} l_j^{(k)} \qquad (5.33)$$

$$u_j^{n+1} = u_j^* + u_j' \qquad (5.34)$$

$$(5.35)$$

11. $n \leftarrow n + 1$; goto step 2 and repeat the process until steady state is reached.

The algorithm developed here is time-accurate, i.e., it can be used to solve unsteady flow problems. However, often such methods are used to solve steady state problems using the false-transient approach, i.e., starting from arbitrary fields and time-marching to the steady-state solutions. In this algorithm, we solve only for the pseudo pressure not for the true pressure. The pseudo pressure is obtained by a rather simple homogeneous Neumann boundary condition instead of the complicated non-homogeneous, and *implicit* boundary condition needed for the true pressure. On staggered grids, the true pressure can be obtained from the pseudo pressure corrections. On collocated grids, however, pressure corrections cannot be used to obtain the pressure, if the pressure boundary conditions are not explicitly implemented. However the divergence free-velocity field is obtained accurately.

5.4   Boundary Conditions

The following types of boundary conditions, generally encountered in fluid-flow problems, can be incorporated into this algorithm.

- No-slip boundary conditions :

$$u_p = 0;$$

- Specified velocity boundary conditions :

$$u_p = u_0;$$

- Free-slip boundary conditions :

$$u_n = 0, \qquad \frac{\partial u_t}{\partial n} = 0;$$

where n and t represents the normal and tangent components.

The boundary conditions for the pressure (p) and pseudo pressure correction (ψ') are

For pressure $\qquad\qquad\qquad \frac{\partial p}{\partial n} = - \left[\frac{Du_n}{Dt} - \frac{1}{Re} \nabla^2 u_n \right]$

For the pseudo pressure correction $\quad \frac{\partial \psi'}{\partial n} = 0.$

It is convenient to use homogeneous boundary conditions, thus pseudo pressure is preferred here to get divergence-free velocity fields. However, to extract correct pressure field it is necessary to use correct boundary conditions for the the pressure obtained from the Navier-Stokes equations as shown above.

At the exit boundary, for steady-state flows the boundary conditions $\frac{\partial u_1}{\partial x_1} = 0, \frac{\partial u_2}{\partial x_1} = 0$, the so-called *fully developed flow* approximation, are used. At the inlet boundary, the inlet velocity is specified. Boundary conditions on solid walls are *no-slip* for velocity, and the pseudo pressure correction boundary conditions are as shown above.

5.5   Results

The calculation procedure is here applied to five steady-state and one unsteady flow problems. Four steady-state test problems are solved on uniform grids, while one (the third) problem shows the efficacy of the method on nonorthogonal grids. The results in all cases are compared with either analytical solutions or with benchmark

solutions reported in the literature. All solutions including the steady-state solutions are obtained by time-marching the unsteady governing equations :

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{5.36}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + = -\frac{\partial \psi}{\partial x} + \frac{1}{Re}\nabla^2 u \tag{5.37}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + = -\frac{\partial \psi}{\partial y} + \frac{1}{Re}\nabla^2 v, \tag{5.38}$$

from the initial conditions

$$u = v = \psi = 0, \qquad \text{at} \quad t = 0.$$

5.5.1 Flow between Concentric Rotating Cylinders on Uniform Grids

The schematic of the problem is shown in Figure 5.1. The inner cylinder is at rest, while the outer cylinder rotates with angular velocity ω. The inner and outer radii of the rotating cylinders are r_1 and r_2 as shown in the figure. The exact solution of this problem is known and can be used to determine the accuracy of the computed solution. The aim is to solve this two-dimensional problem over the square region shown in the Figure 5.1. The boundary conditions are of Dirichlet type for velocities and prescribed using the analytical solution. The homogeneous Neumann boundary condition has been used for the pseudo pressure. The analytical solution, which is independent of Reynolds number, is :

$$u = \frac{(2r - 2/r)y}{3r} \tag{5.39}$$

$$v = -\frac{(2r - 2/r)x}{3r} \tag{5.40}$$

$$p = \frac{2(r^2 - 1/r^2) - 8ln(r)}{9} \tag{5.41}$$

where u, v, p, are the two components of velocity, and pressure, non-dimensionalized by $2r$, ω, and $\rho(2r\omega)^2$, respectively.
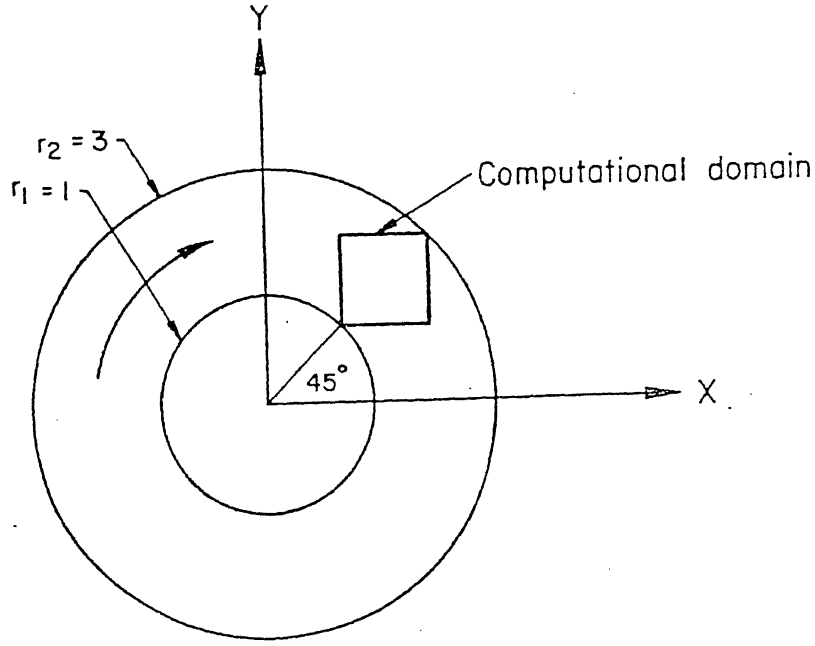
Figure 5.1: Schematic diagram of test problem 1

The computational domain is divided into $N \times N$ grid points. The results are obtained for four grids, $N=$ 11, 21, 31, and 41. The Reynolds number ($R_e = \rho(2r_1\omega)r_1/\mu$) is varied from 1 to 1000 and the radius ratio (r_2/r_1) is kept fixed at 2.

The variation of the absolute percentage error in u at the midpoint of the computational domain, and the average percentage of absolute error over the domain, with Reynolds number and grid size are tabulated in Table 5.1 and 5.2, respectively. It can be seen that the method is second-order accurate : doubling the grid (for fixed Reynolds number) reduces the error by at least a factor of 4. This is valid for the entire range of Reynolds numbers from the diffusive (Re=1) to the convective (Re=1000) limits. The results obtained using OCV scheme are compared with unequal-order method of Baliga and Patankar[84] (see Tables 5.1 and 5.2) and equal-order method of Prakash and Patankar[85] for 21 × 21 grid (see Tables 5.1 and 5.2). It can be observed that the results obtained using OCV approach compare well with those of these methods. Also, it is likely that these other methods, being finite-element based, would require considerably more computational effort for the same grid.

Table 5.1: Percentage Error in u at the center of domain

	Grid	$R_e = 1$	$R_e = 10$	$R_e = 100$	$R_e = 1000$
OCV	11 × 11	0.1171	0.1442	0.4169	0.6919
OCV	21 × 21	0.0289	0.0321	0.0431	0.0718
OCV	31 × 31	0.0128	0.0125	0.0199	0.0470
OCV	41 × 41	0.0071	0.0067	0.0109	0.0324
Baliga and Patankar[84]	21 × 21	0.020	0.034	0.264	0.287
Prakash and Patankar[86]	21 × 21	0.00404	0.0241	0.193	0.149

Table 5.2: Average Percentage Error in u over the domain

	Grid	$R_e = 1$	$R_e = 10$	$R_e = 100$	$R_e = 1000$
OCV	11 × 11	0.1222	0.1584	0.4611	0.5724
OCV	21 × 21	0.0296	0.0320	0.0508	0.0757
OCV	31 × 31	0.0131	0.0140	0.0224	0.0424
OCV	41 × 41	0.0075	0.0079	0.0134	0.0299
Baliga and Patankar[84]	21 × 21	0.034	0.060	0.407	0.594
Prakash and Patankar[86]	21 × 21	0.00846	0.0162	0.114	0.166

Figures 5.2, 5.3 and 5.4 shows a comparison of the analytical and computed solutions for the u-velocity, v-velocity and pressure along the main diagonal of the computational domain for Reynolds number from 1 to 1000. It can be observed that the computed solutions are in good agreement with the analytical solutions. Figures 5.5-5.10 shows the computed solutions for u-velocity, v-velocity and pressure along the main diagonal for three different grids 11×11, 21×21 and 41×41 for Re= 10 and 1000. The pressure computed is actually the pseudo pressure, *i.e.* it is obtained by using the homogeneous Neumann boundary conditions, but seems to give good results except at the boundary points.
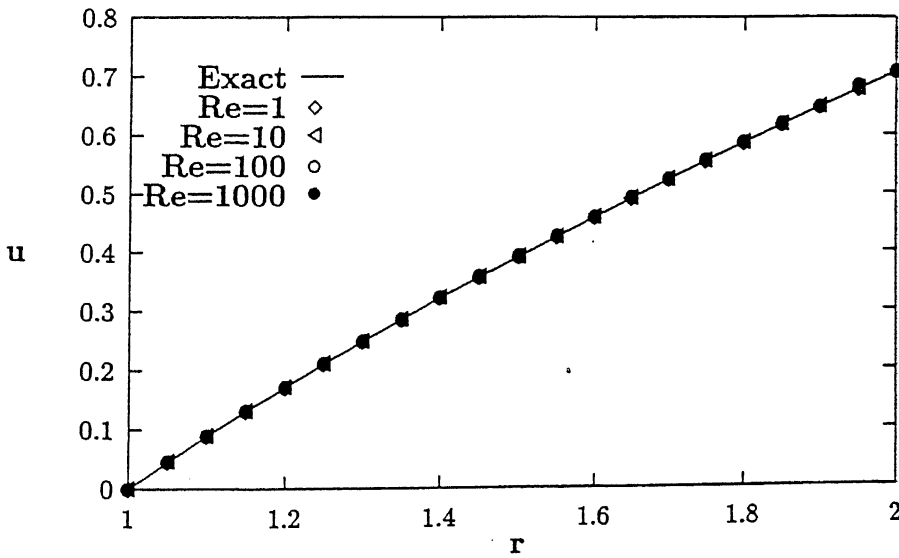
Figure 5.2: Variation of u-velocity along main diagonal (Grid=21 × 21).
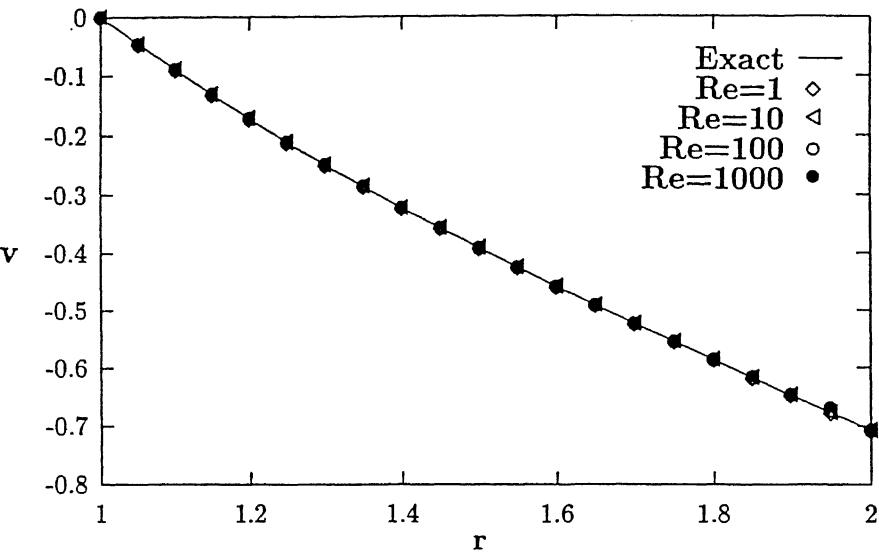
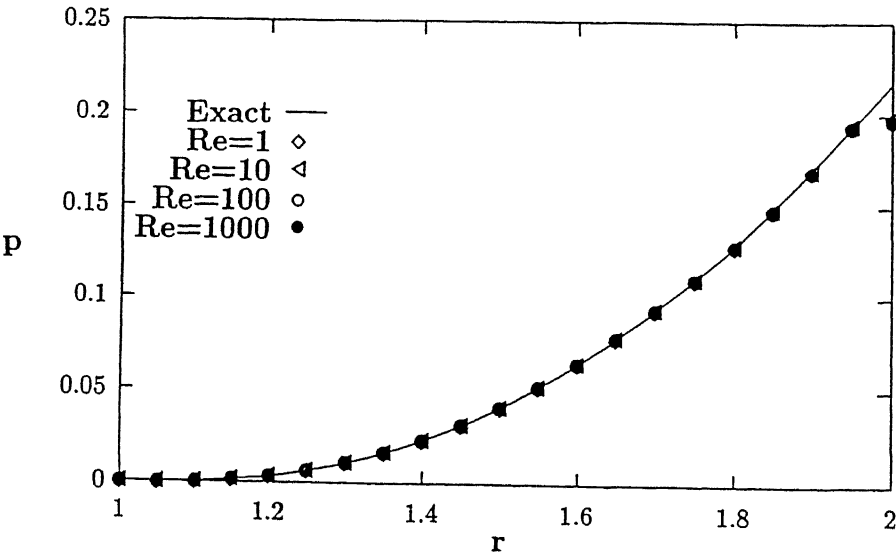Figure 5.3: Variation of v-velocity along main diagonal (Grid=21 × 21).

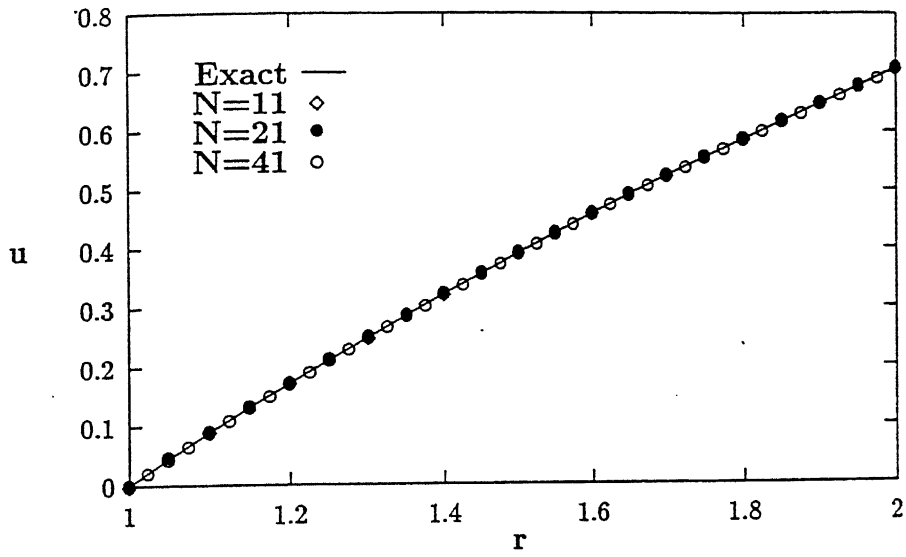Figure 5.4: Variation of pressure along main diagonal (Grid=21 × 21).

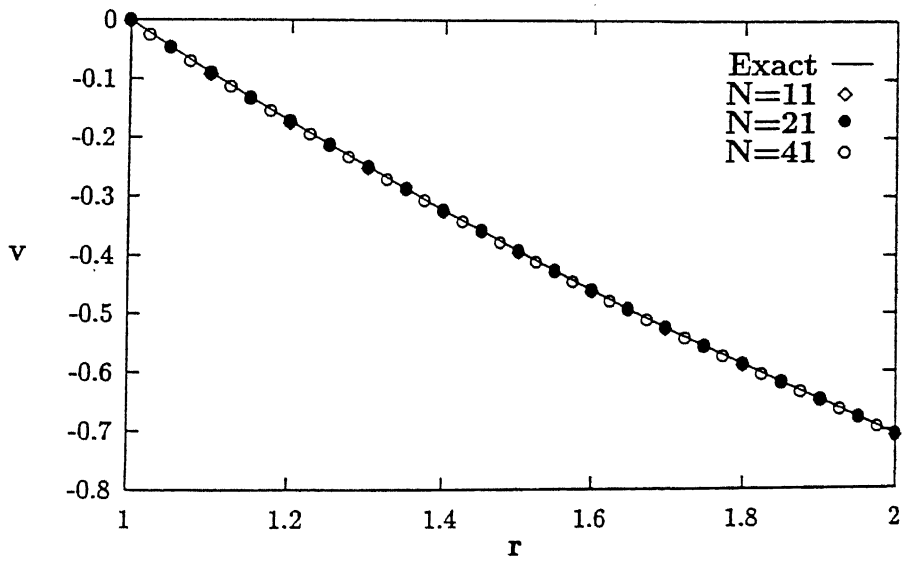Figure 5.5: Variation of u-velocity along main diagonal for R_e=10.

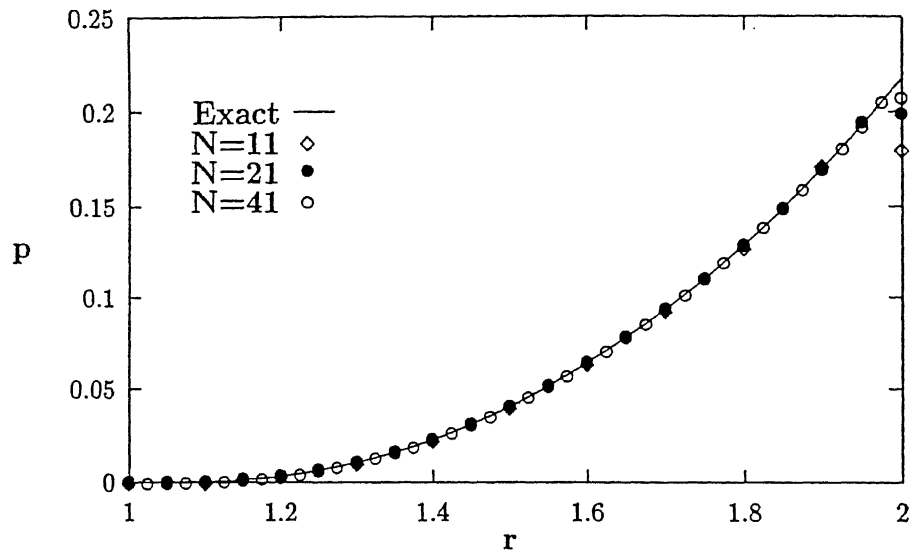Figure 5.6: Variation of v-velocity along main diagonal for R_e=10.

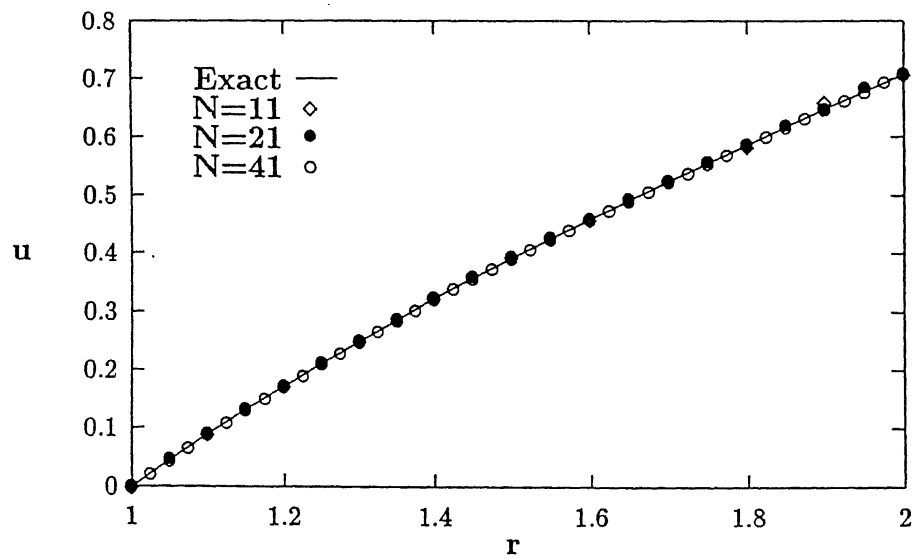Figure 5.7: Variation of pressure along main diagonal for R_e=10.

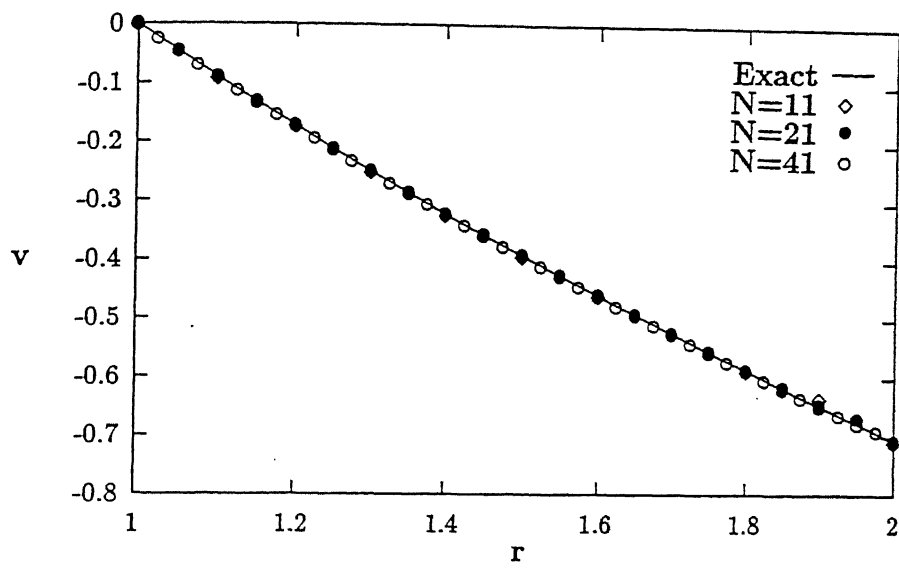Figure 5.8: Variation of u-velocity along main diagonal for R_e=1000.

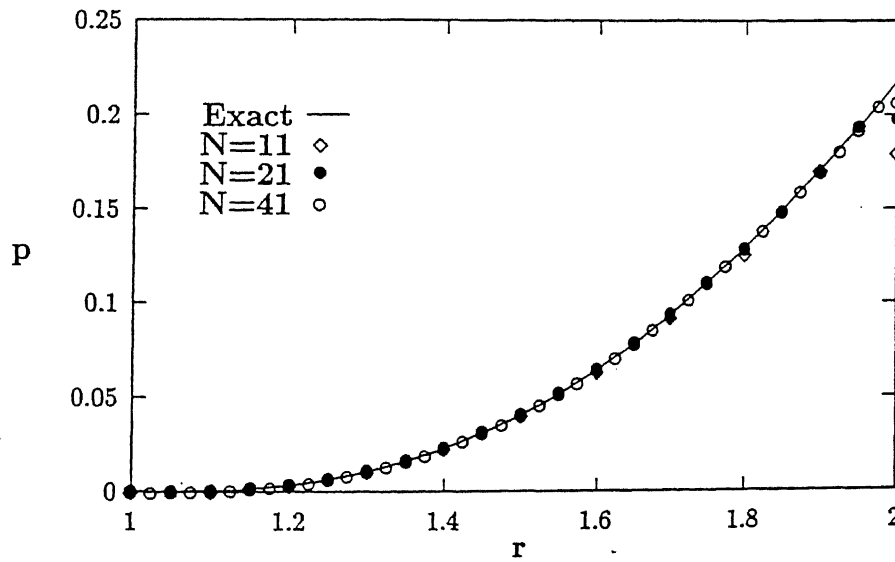Figure 5.9: Variation of v-velocity along main diagonal for R_e=1000.

Figure 5.10: Variation of pressure along main diagonal for R_e=1000.

5.5.2   Flow in a Driven Square Cavity

The problem considered here [Figure 5.11] is that of a two-dimensional square cavity of unit dimensions. The motion of an incompressible viscous fluid in the cavity is induced by the motion of the lid. Steady-state two-dimensional laminar flow in a driven square cavity is a good benchmark problem because it offers a deceptively simple model on which numerical techniques may examined and very accurate numerical results are available for comparison [Ghia et al[126]].

The nature of the vortex formed in the cavity depends on the aspect ratio (Cavity height to width ratio) as well as the Reynolds number ($Re \equiv \frac{Ud}{\nu}$), where U, d, ν are the velocity of the top lid, width of the cavity (assumed to be unity), and the kinematic viscosity, respectively.
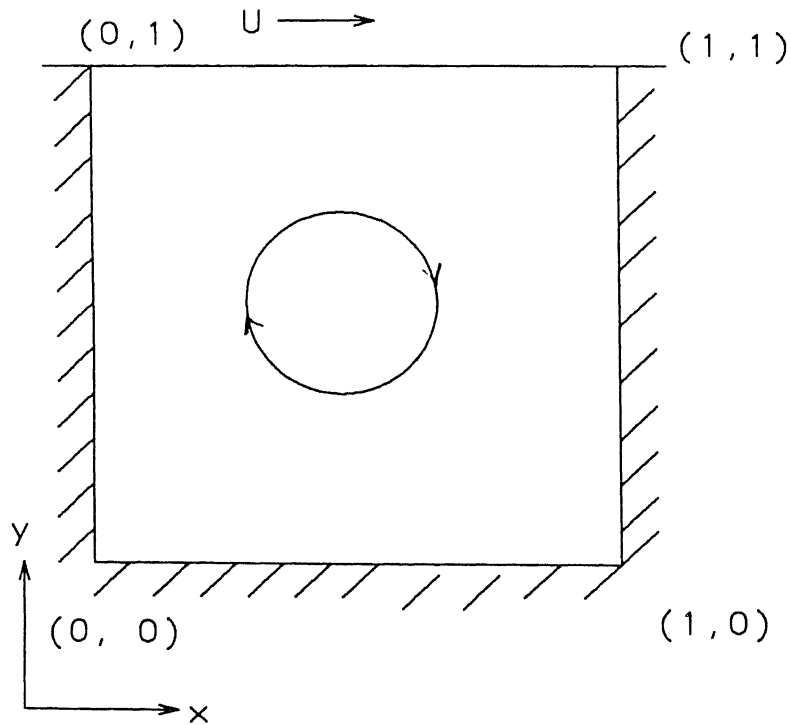

Figure 5.11: Schematic diagram of square cavity.

The schematic of the problem is illustrated in Figure 5.11. The boundary conditions for the velocity components are given by (for $t > 0$),

$$
\begin{aligned}
u &= v = 0 \quad \text{at} \quad x = 0, x = 1 \\
u &= v = 0 \quad \text{at} \quad y = 0 \\
u &= 1, v = 0 \quad \text{at} \quad y = 1
\end{aligned}
$$

and $\frac{\partial \psi'}{\partial n} = 0$ on the solid boundaries. The velocities have been nondimensionalized with U, and all lengths the cavity width d.

The computational domain is divided into $N \times N$ uniform grids. The results have been obtained on grids corresponding to $N=$ 41, 61, and 81, for Reynolds numbers of 100, 400 and 1000. Figures 5.12 and 5.13 show the plot of the profiles of horizontal velocity along the vertical centerline and vertical velocity along the horizontal centerline of the cavity for $Re=$ 100. Figure 5.14 and 5.15 show the results for $Re=$400. The plots also show the finite-difference results of Ghia et al.[126] on a finer grid of 129 $\times$ 129. It can be seen that the results obtained by OCV are in close agreement with the reference solution. The vectors plot for Reynolds number corresponding to 100, 400 and 1000 are shown in Figures 5.16-5.18. It can be observed that for aspect ratio of unity and relatively low Reynolds numbers, most of the strength of the vortex is concentrated in the upper portion of the cavity. As the Reynolds number increases, the vortex center moves downstream. With further increase in Reynolds number, the vortex center moves down and towards the center of the cavity. In figure 5.18, the magnified view of the velocity vectors at the bottom clearly shows the presence of secondary circulation. It can be clearly seen that the extent of the secondary circulation increases with the increase in the Reynolds number as expected. Figures 5.19-5.21 show streamlines for the same computational solutions.
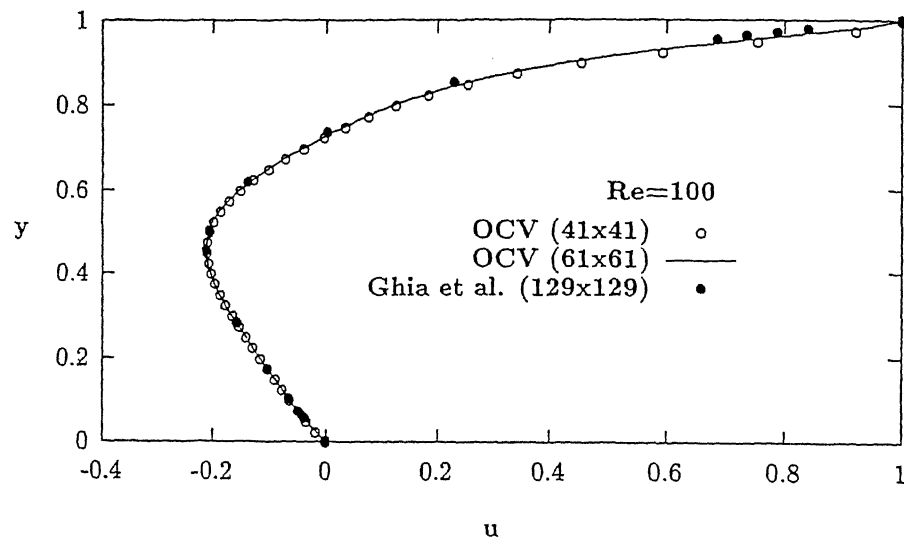
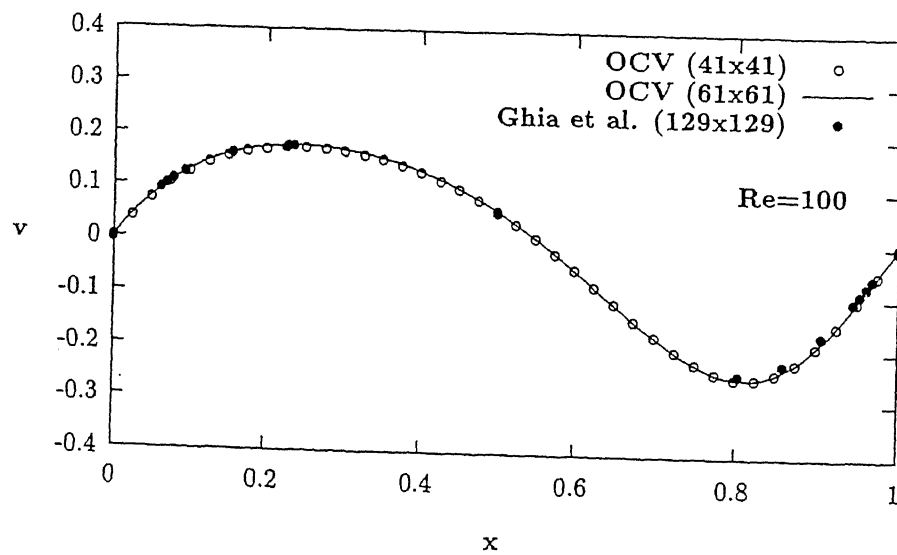Figure 5.12: Velocity profiles on vertical centerline of a square cavity, Re = 100

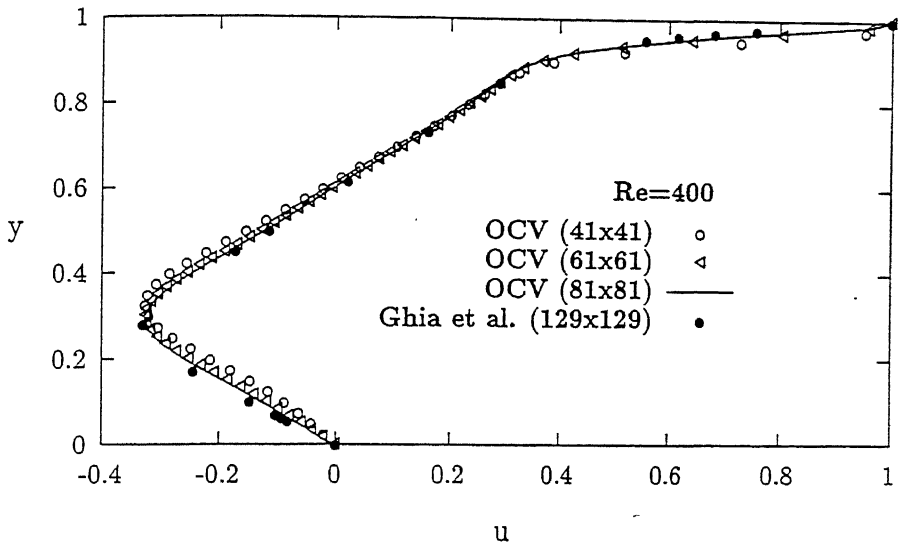Figure 5.13: Velocity profiles on horizontal centerline of a square cavity, Re = 100

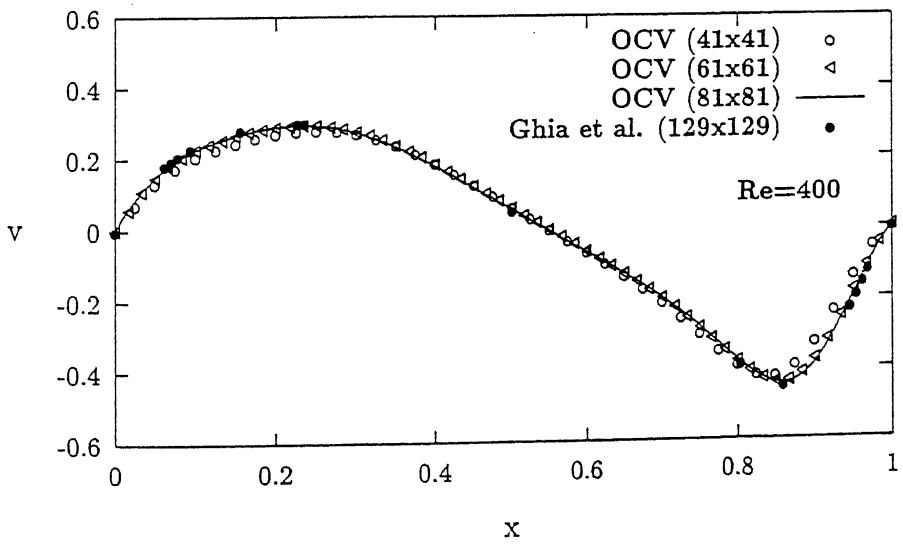Figure 5.14: Velocity profiles on vertical centerline of a square cavity, Re = 400

Figure 5.15: Velocity profiles on horizontal centerline of a square cavity, Re = 400

Figure 5.16: Velocity vectors in a lid driven cavity for R_e=400 (41 × 41 grid)
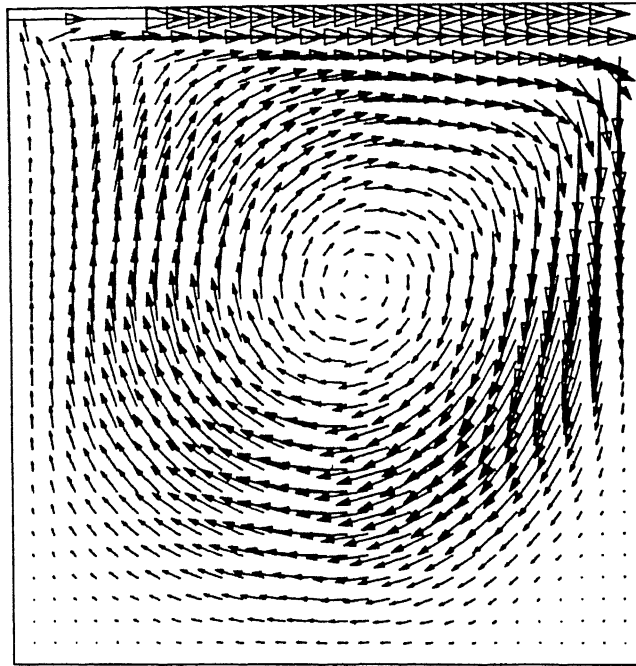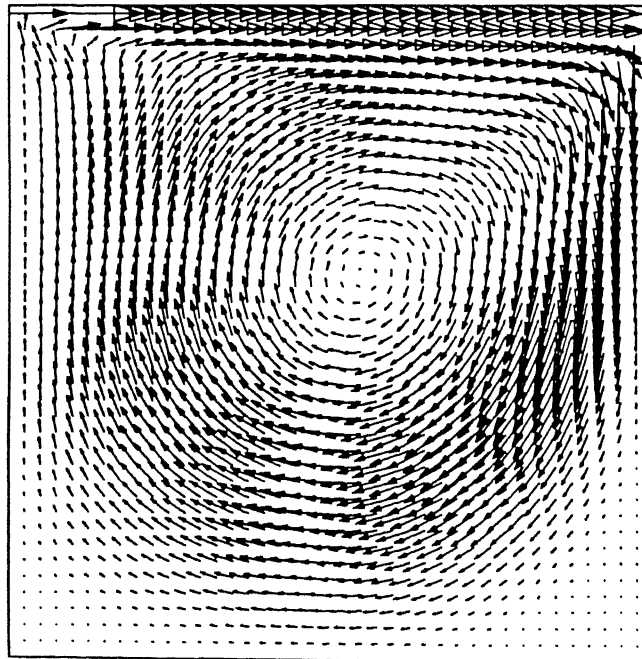

Figure 5.17: Velocity vectors in a lid driven cavity for R_e=400 (81 × 81 grid)

Figure 5.18: (a) Velocity vectors in a lid driven cavity for R_e=1000 (81 × 81 grid)
(b) Magnified view of the right bottom corner of the cavity

Figure 5.19: Streamlines in a lid driven cavity for R_e=100 (41 × 41 grid)

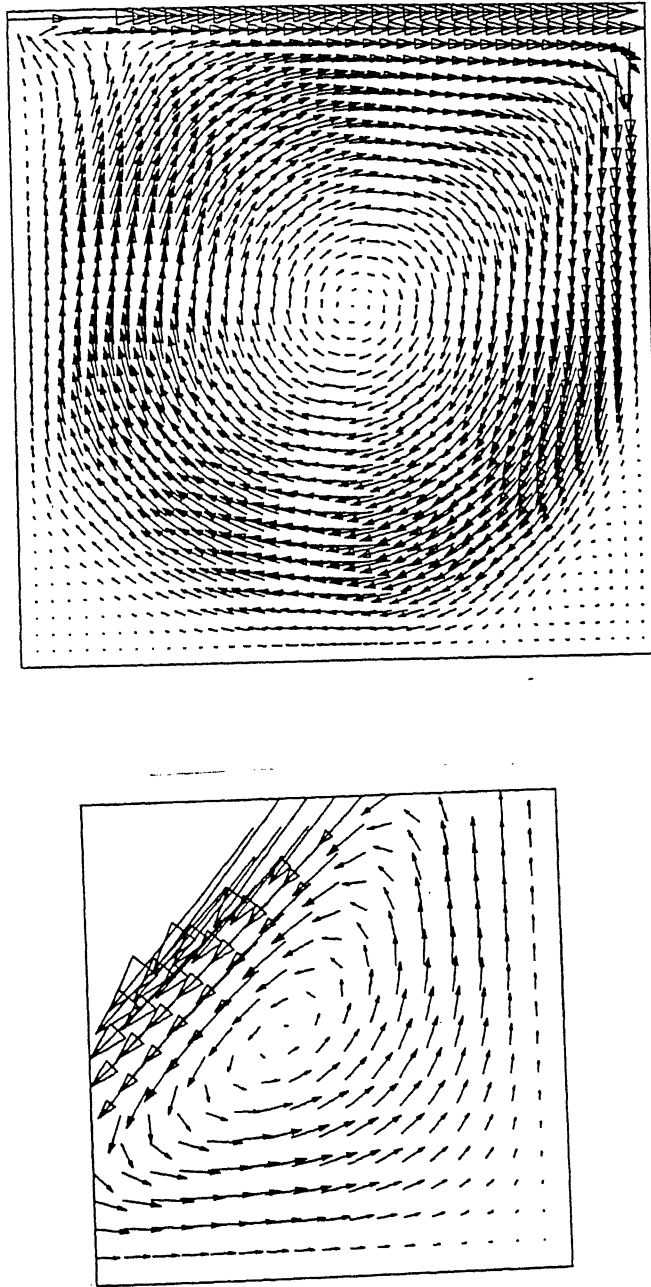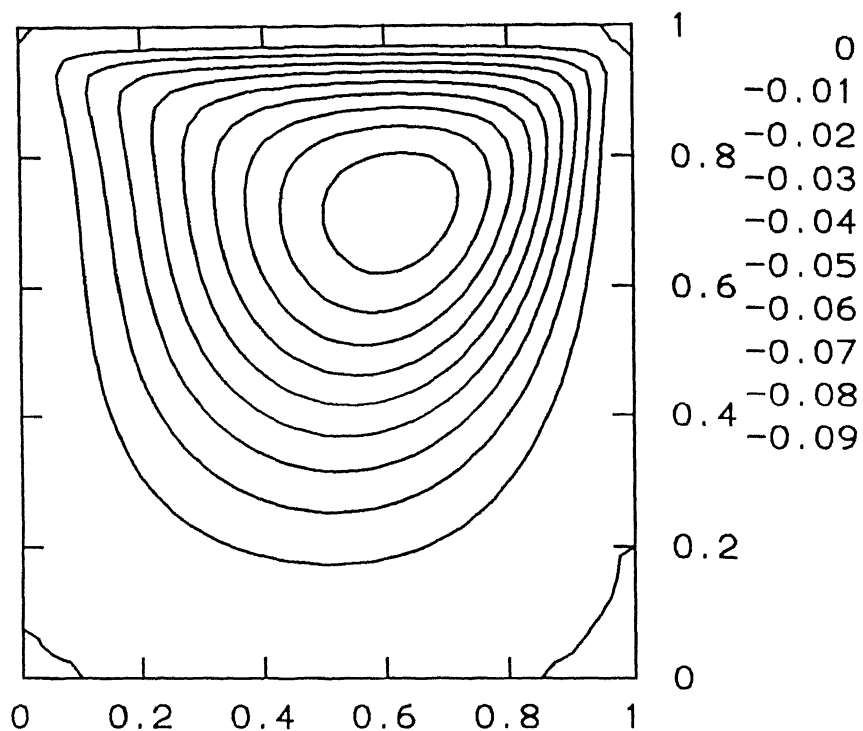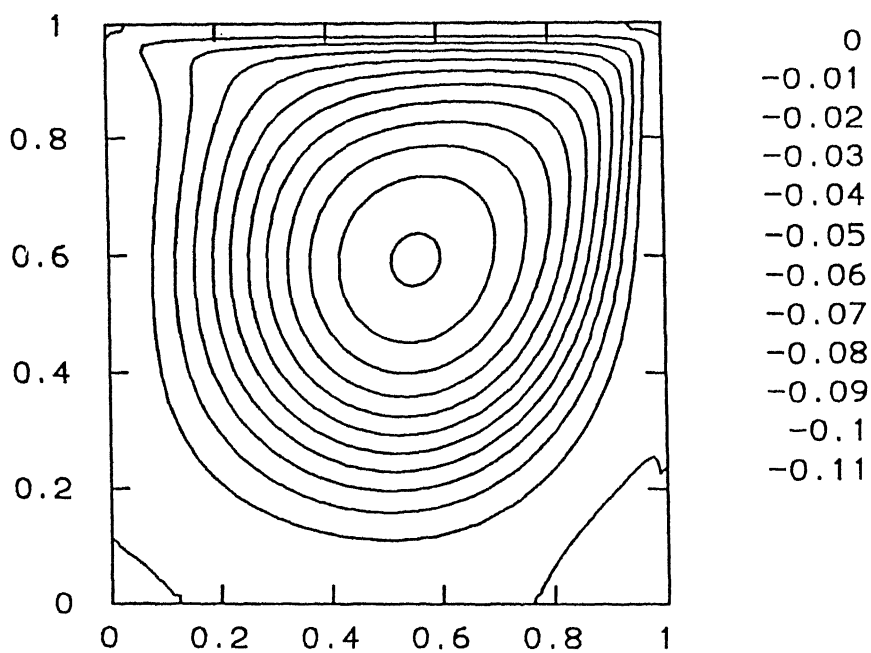

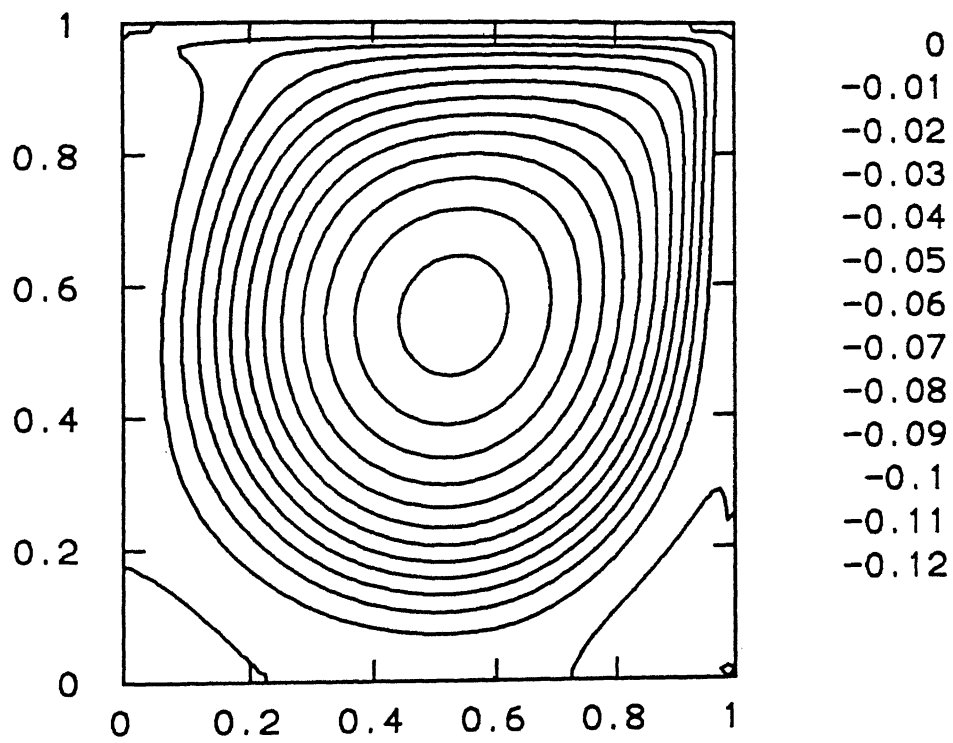Figure 5.20: Streamlines in a lid driven cavity for R_e=400 (81 × 81 grid)

Figure 5.21: Streamlines in a lid driven cavity for R_e=1000 (81 × 81 grid)

5.5.3   Flow between Concentric Rotating Cylinders on Nonorthogonal Grids

This problem is designed to test the applicability of the method to nonorthogonal grids. The details and exact solution of this problem is same as those of Test Problem 1. The only difference lies is the interior grid points are randomly perturbed from their original uniform grid positions to a maximum extent of 5, 10 and 20 percent of the average grid distance in both the directions. Figure 5.22 shows the 20 per cent distorted grid.
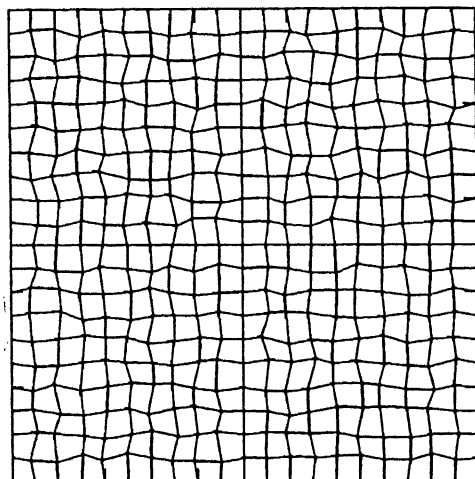
Figure 5.22: Distorted grid for test problem 3.

The results for the above three distorted grids are presented in the Tables 5.3, 5.4, and 5.5 for N= 21, 31 and 41. The u velocity profiles along main diagonal of the domain is shown in 5.23 and 5.24 for the 10 per cent and 20 percent distorted grids, respectively. Figure 5.25 and 5.26 show the corresponding v-profile. There is little effect on the solution due to nonorthogonality of the grid. It is also observed that even on moderately nonorthogonal grids with 20 percent grid distortion, results obtained using OCV method is comparable to Baliga and Patankar[84] and Prakash and Patankar[86] (Table 5.1 and 5.2). The pressure plots are shown in Figures 5.27, 5.28. It can be seen that the non-orthogonality causes pressure solutions to differ from the analytical solutions. However, the magnitude of error decreases for higher Reynolds numbers where the true pressure boundary condition is close to the homogeneous Neumann condition used in these computations.

Table 5.3: Average Percentage Error in u over the domain : OCV Method (5 percent grid-distortion)

Grid	$R_e = 1$	$R_e = 10$	$R_e = 100$	$R_e = 1000$
21 × 21	0.0296	0.0340	0.0827	0.1409
31 × 31	0.0132	0.0144	0.0290	0.0616
41 × 41	0.0075	0.0079	0.0140	0.0339

Table 5.4: Average Percentage Error in u over the domain : OCV Method (10 percent grid-distortion)

Grid	$R_e = 1$	$R_e = 10$	$R_e = 100$	$R_e = 1000$
21 × 21	0.0296	0.0338	0.0782	0.1979
31 × 31	0.0135	0.0142	0.0313	0.1190
41 × 41	0.0077	0.0078	0.0209	0.0866

Table 5.5: Average Percentage Error in u over the domain : OCV Method (20 percent grid-distortion)

Grid	$R_e = 1$	$R_e = 10$	$R_e = 100$	$R_e = 1000$
21 × 21	0.0291	0.0205	0.1130	0.3348
31 × 31	0.0163	0.0165	0.0570	0.2426
41 × 41	0.0091	0.0092	0.0455	0.1383

Figure 5.23: Variation of u-velocity along main diagonal on 10 percent distorted grid (Grid=21 × 21).

Figure 5.24: Variation of u-velocity along main diagonal on 20 percent distorted grid (Grid=21 × 21).

Figure 5.25: Variation of v-velocity along main diagonal on 10 percent distorted grid (Grid=21 × 21).

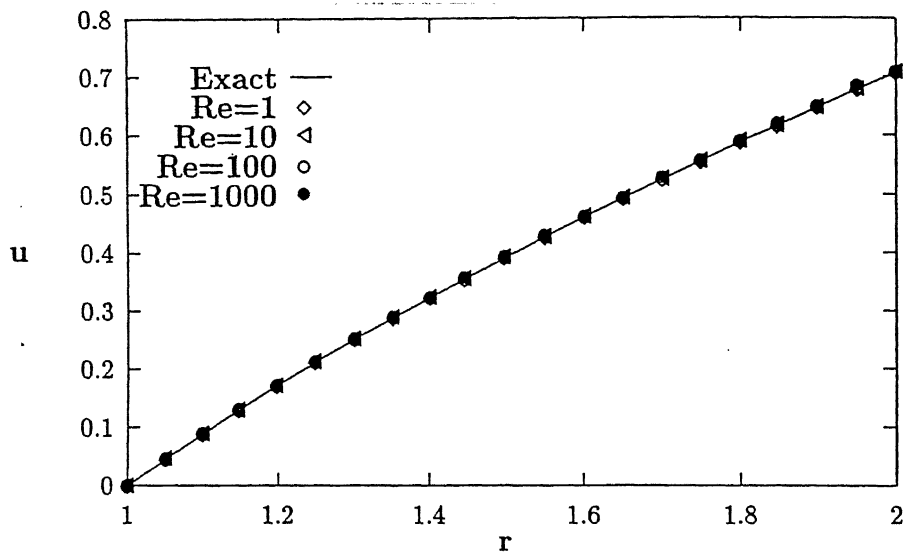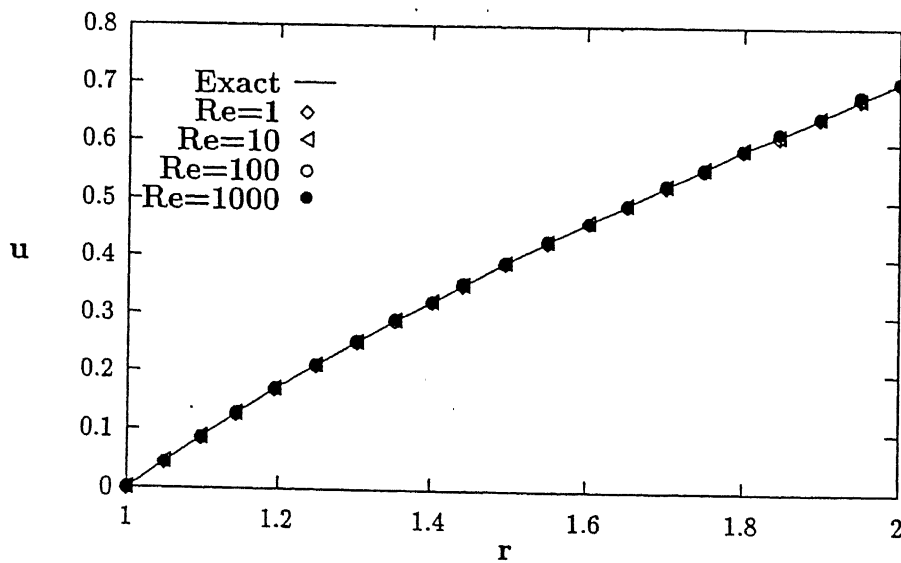Figure 5.26: Variation of v-velocity along main diagonal on 20 percent distorted grid (Grid=21 × 21).

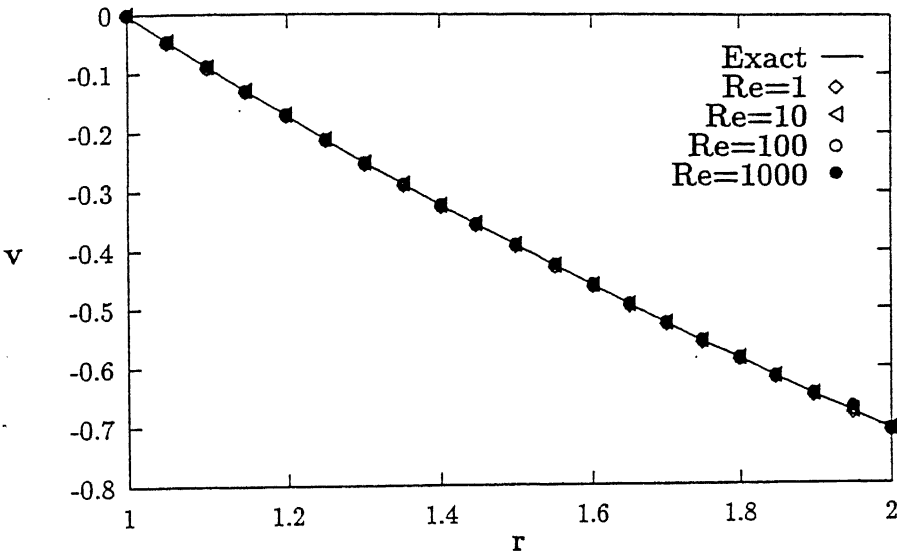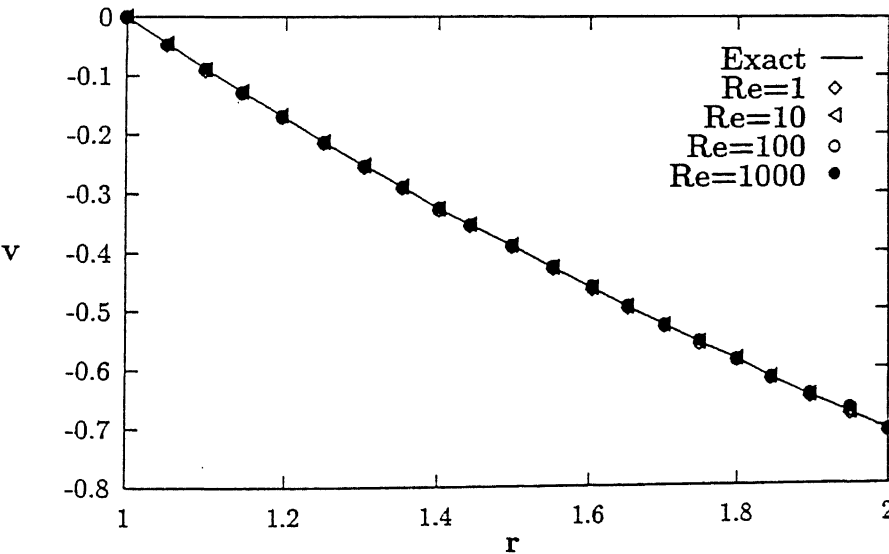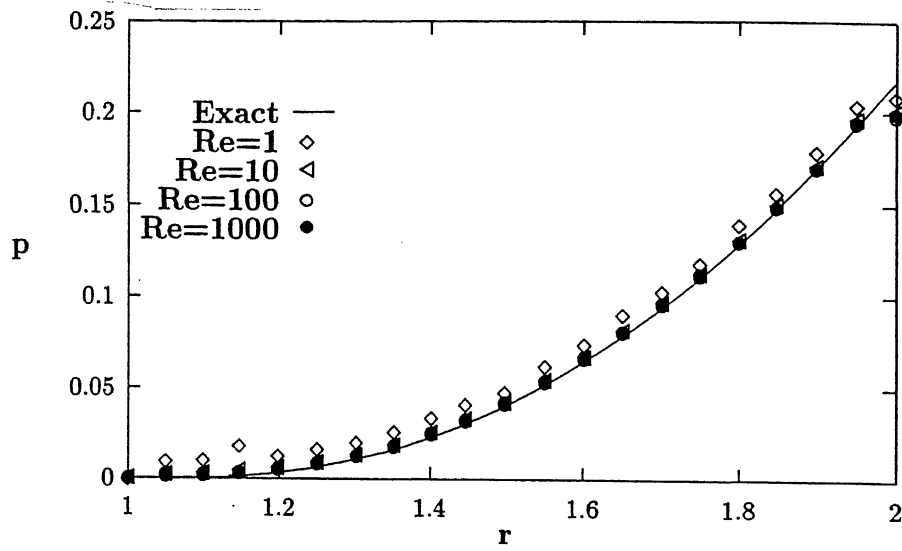Figure 5.27: Variation of pressure along main diagonal on 10 percent distorted grid (Grid=21 × 21).
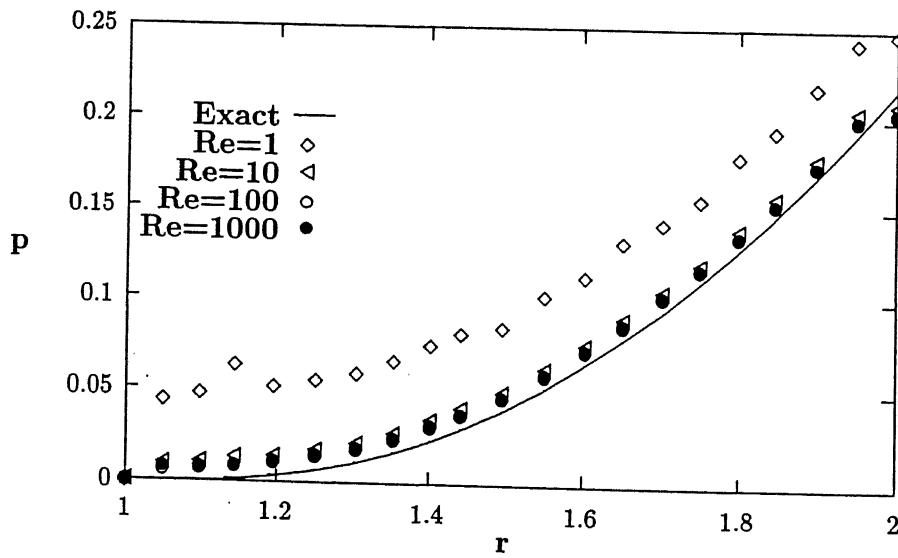

Figure 5.28: Variation of pressure along main diagonal on 20 percent distorted grid (Grid=21 × 21).

5.5.4 Flow through a Channel with Sudden Expansion

The two-dimensional laminar flow over a sudden expansion/backward-facing step in a channel provides an excellent test case for the accuracy of numerical scheme because of the dependence of the reattachment length (Lr) on the Reynolds number. Excessive numerical diffusion or smoothing in order to get stability will result in failure to predict correct reattachment length. The schematic diagram for this problem is shown in Figure 5.29. The non-dimensional length of the channel (L) is 6 units and the non-dimensional height at the channel inlet (H1) and outlet (H2) are one and two units, respectively and the step is located at a downstream distance Ls of 2 units. The following boundary conditions have been used for the velocity components in the present computations :

$$u = 240y(1-y), \qquad x = 0, \qquad 0 \leq y \leq H1; \qquad v = 0, \qquad x = 0, \qquad 0 \leq y \leq H1;$$

$$u = v = 0, \qquad y = 0, \qquad 0 \leq x \leq L; \qquad u = v = 0, \qquad y = H1, \qquad 0 \leq x \leq Ls;$$

$$u = v = 0, \qquad x = Ls, \qquad H1 \leq y \leq H2; \qquad u = v = 0, \qquad y = H2, \qquad Ls \leq x \leq L;$$

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} = 0, \qquad x = L, 0 \leq y \leq H2$$

The computational domain, the 2×6 box enclosing the flow, is divided into a uniform grid. Three grids corresponding to 91×41, 121×61 and 181×61 grid points are used. The Reynolds number (based on the maximum velocity at inlet, $u_{max} = 60.0$) used for the computations is 60. The u-velocity profile at $x=3$ units from the inlet section and coefficient of friction $(C_f = \frac{\tau_{wall}}{(1/2)\rho u_{max}^2})$ along the top wall in the sudden expansion zone are presented in Figures 5.30 and 5.31, respectively. These results are in close agreement with the results reported by Reddy et al.[127] for $R_e=60$ with a finite element method. The reattachment length, L_r, (where $C_f = 0$) obtained with the 91×41 and 181×61 grids are found to be 2.4060 and 2.4135, respectively whereas Reddy et al.[127] obtained a value of 2.42093 by using a FRONTAL solver. The velocity vector plot and streamlines are presented in Figures 5.32 and 5.33, respectively.
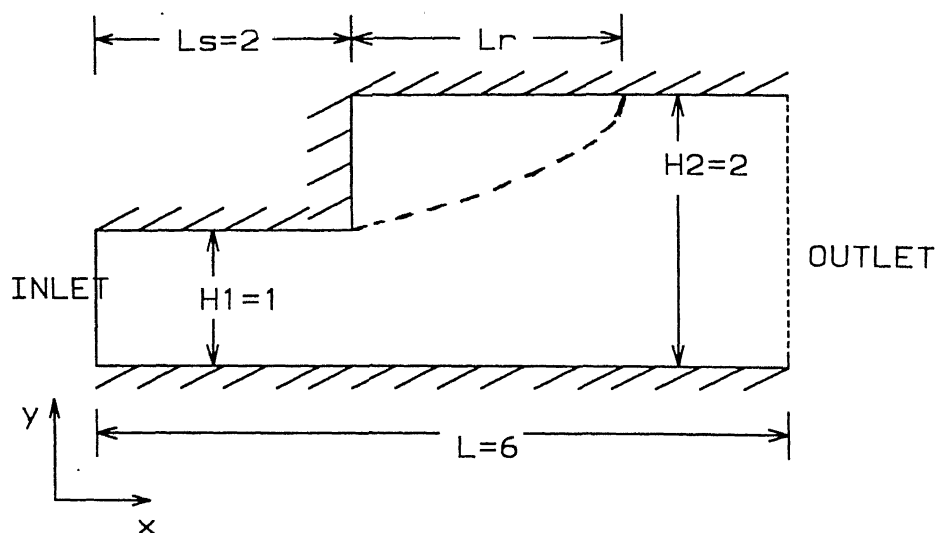
Figure 5.29: Schematic diagram of a channel with sudden-expansion
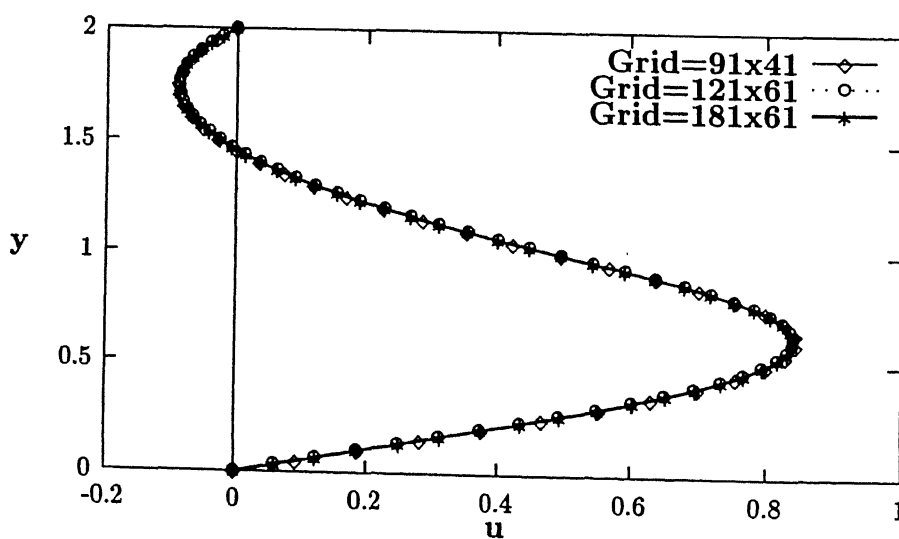

Figure 5.30: The u-velocity profile at $x=3.0$
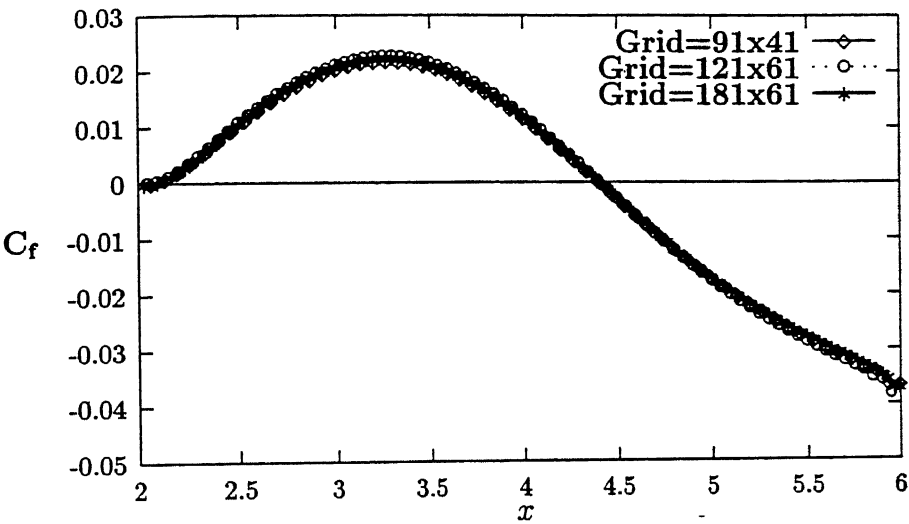
Figure 5.31: The variation of coefficient of friction along the top wall downstream from sudden expansion
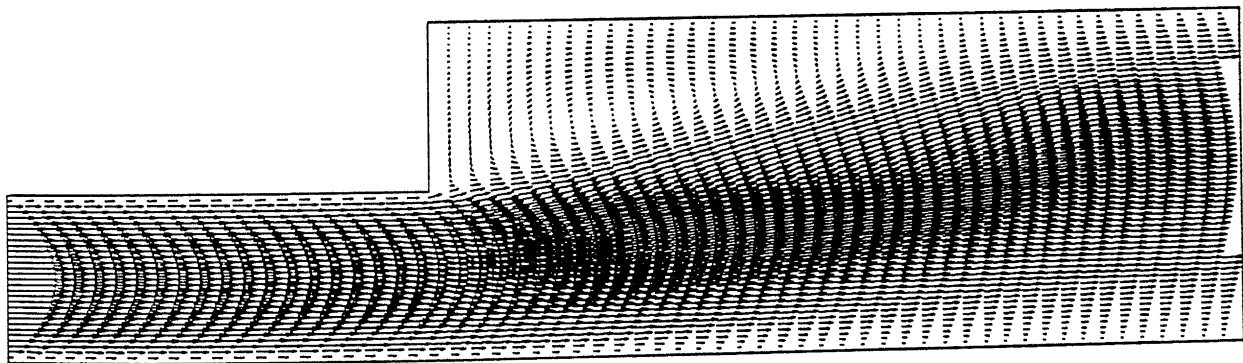
Figure 5.32: Velocity vector plot for sudden expansion test case (R_e=60)
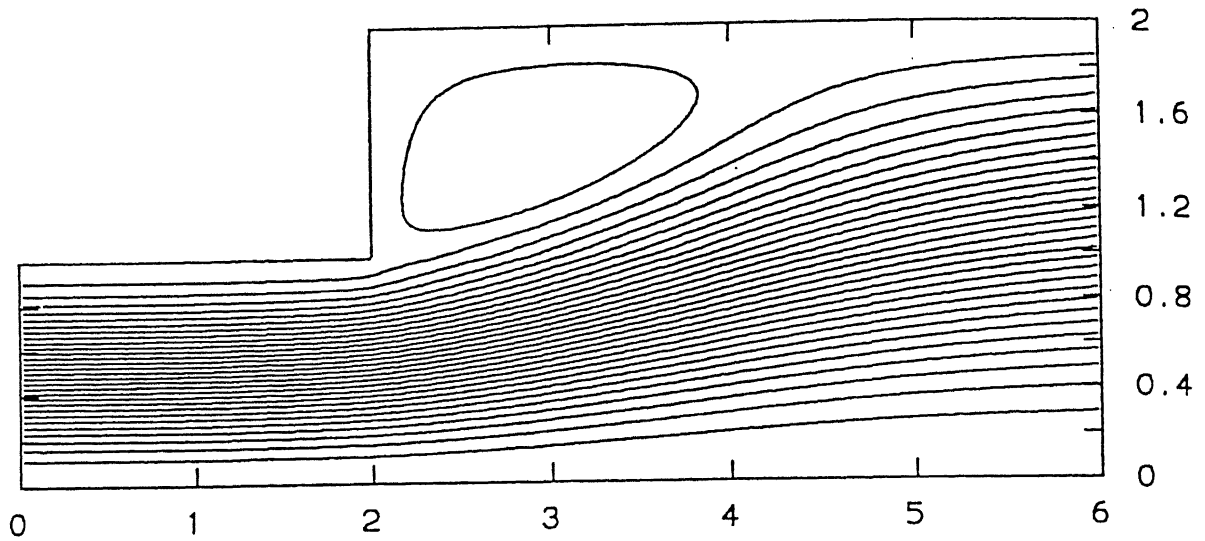
Figure 5.33: Streamlines plot for sudden expansion test case (R_e=60)

5.5.5   Flow through a Channel with Backward-Facing step

This problem is essentially the same as the previous one, except that the details are changed so as to compare with the results of other researchers. The geometry and boundary conditions are shown in Figure 5.34. The step height is half the channel height and the step is located at $L_s = 0$, from the inlet. The Reynolds number is defined by $R_e = \frac{u_{avg}H}{\nu}$. At the inflow boundary, located at the step, a parabolic profile $u(y) = 6y(1-y)$ is prescribed. This produces a maximum inflow velocity of $u_{max} = 1.5$ and average inflow velocity $u_{avg} = 1.0$. The experimental results for this particular flow configuration have been given by Armaly et al.[128] and numerical results can be found in the works of Kim and Moin[129], Thompson and Ferziger[130] and Gartling[131].

The steady state solutions for this problem have been obtained for Reynolds number values of 133, 267, 400, and 600. The number of grid points used in the computations, length of the channel and the calculated reattachment length (normalized by the step height) have been presented in Table 5.6. It can be seen that the predicted reattachment lengths (corresponding to $C_f = 0$) are very close to the values reported by Thompson and Ferziger[130] using a 256 × 64 and 512 × 128 grid points as shown in Table5.6.

H/h=2
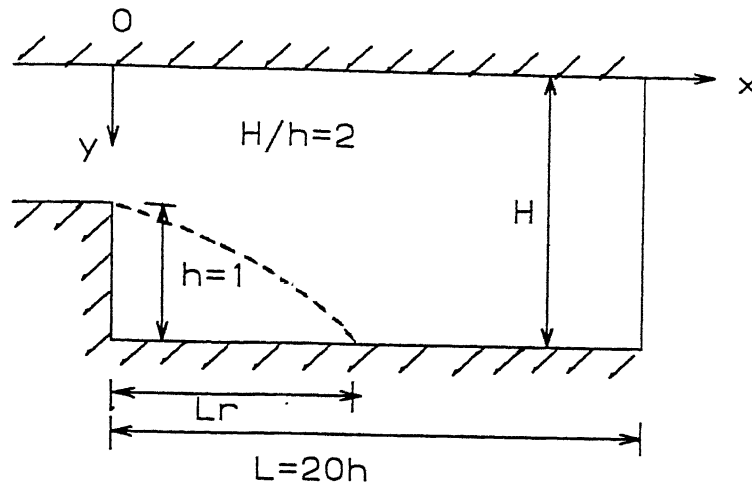
Figure 5.34: Schematic diagram of a channel with Backward   facing step

Table 5.6: Reattachment length (Lr) Vs. Re for backward facing step

Method	Re	Grid	L	Lr
OCV	133	121 × 61	20.0	4.0
OCV	267	121 × 61	20.0	6.6
OCV	400	121 × 61	20.0	8.7
OCV	600	121 × 61	20.0	10.7
Thompson and Ferziger[130]	133	256× 64	12.0	4.0
Thompson and Ferziger[130]	267	256× 64	20.0	6.5
Thompson and Ferziger[130]	400	256× 64	27.0	8.5
Thompson and Ferziger[130]	600	256× 64	36.0	10.1
Thompson and Ferziger[130]	400	512× 128	27.0	8.7
Thompson and Ferziger[130]	600	512× 128	36.0	10.8

The streamlines plots for various Reynolds numbers are shown in Figures 5.35-5.38. The size of the recirculation bubbles increases, as shown in the streamlines plots, with the Reynolds number. For larger Reynolds numbers, a secondary recirculation bubble is also observed on the top wall (see Figure 5.38). This is typical of the observations of the other researchers. The results obtained demonstrate that the the OCV method is quite effective for recirculating incompressible fluid flow predictions.
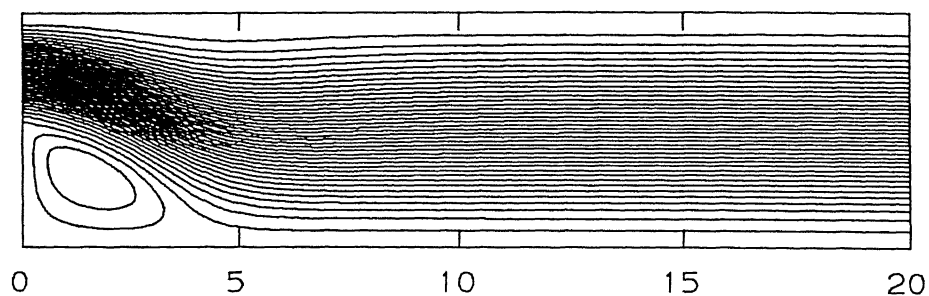
Figure 5.35: Streamlines plot for Backward-facing step test case (R_e=133)
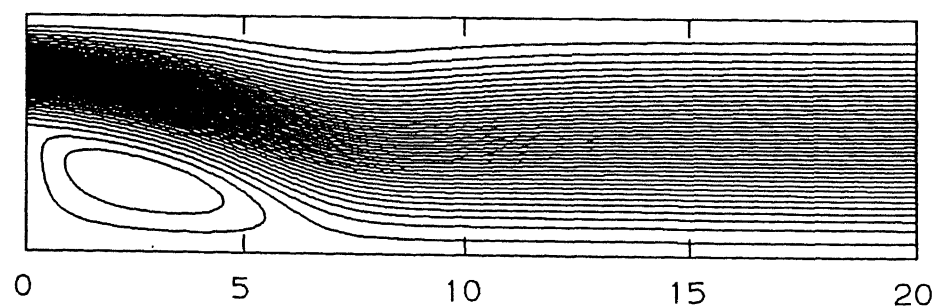
Figure 5.36: Streamlines plot for Backward-facing step test case (R_e=267)
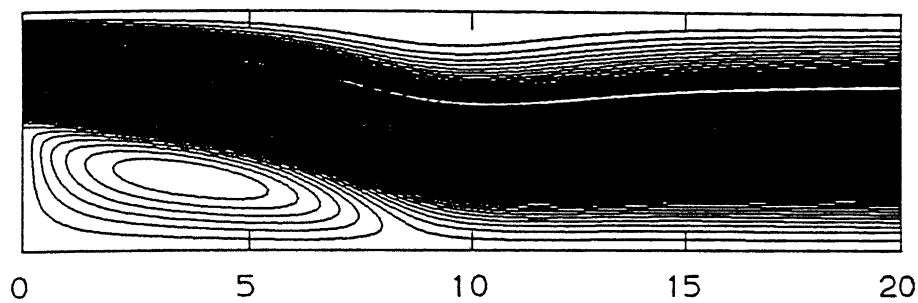
Figure 5.37: Streamlines plot for Backward-facing step test case (R_e=400)
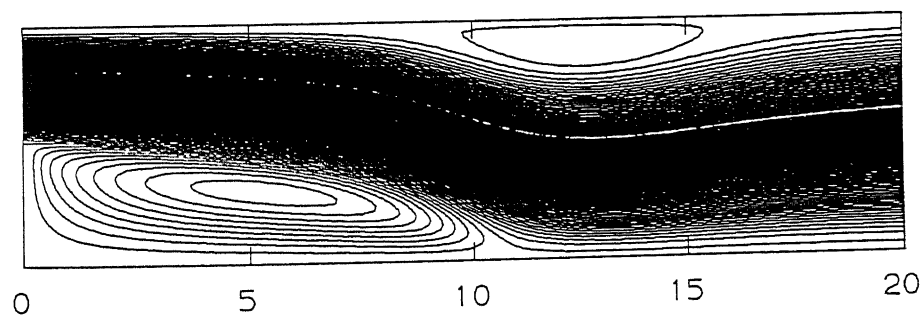
Figure 5.38: Streamlines plot for Backward-facing step test case (R_e=600)

5.5.6   Confined flow around square cylinder

We now consider the unsteady flow around a square cylinder placed in a channel. This problem has been considered by a number of researchers[132-138]. The definition sketch for the problem is given in Figure 5.39. The relevant independent parameters are the blockage-ratio (B/H) and the Reynolds numbers, here based on the average velocity (Re=$\frac{U_{av}B}{\nu}$). At low Reynolds number this problem can have steady-state solutions. However, beyond a critical Reynolds number only unsteady solutions are possible which involve alternate shedding of vortices from upper and lower downstream corners of the obstacle. The dependent variable of interest is the non-dimensional vortex shedding frequency, the Strouhal number $St \equiv \frac{fB}{U_{av}}$, where f is the frequency of the vortex shedding, and its dependence on the blockage ratio and Reynolds number. Also of interest are the critical Reynolds number, and its dependence on the blockage ratio, and the time-variation of the lift and drag coefficients, C_L and C_D, which relate to the vortex shedding in the wake of the cylinder.
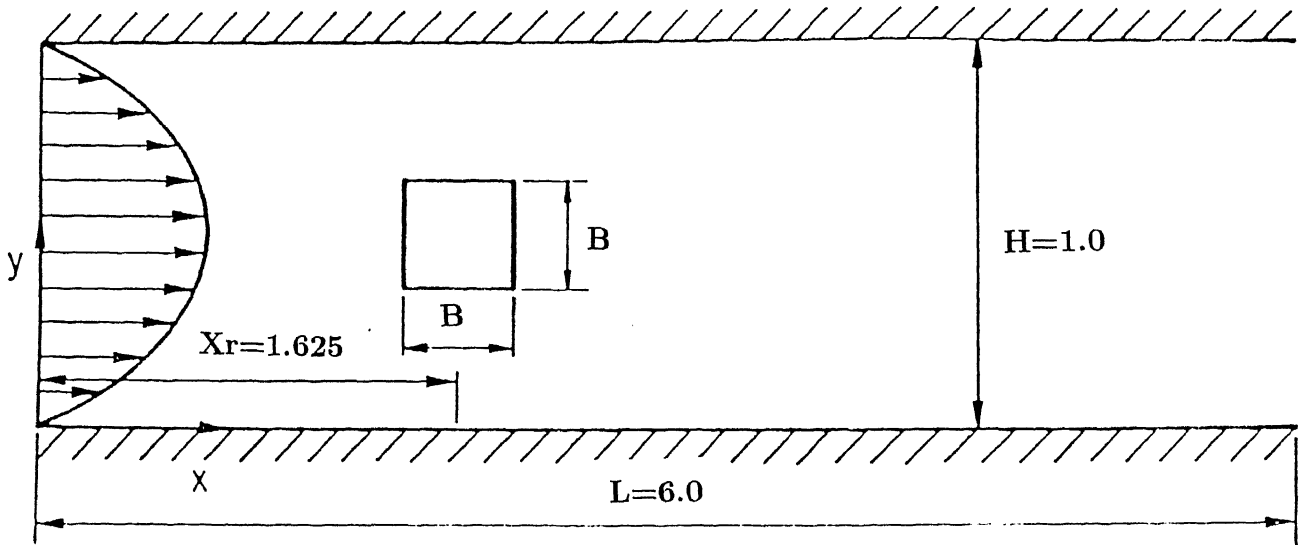
Figure 5.39: Schematic diagram of Test Problem 6

The attempt here, however, is not to repeat the extensive study done on this problem by others (e.g. Davis et al[132]   ). Our purpose is to validate the OCV

algorithm on this unsteady problem and we do computations for only a limited range of parameters in order to establish that the results obtained are comparable with those reported in the literature. So, our computations are confined to cases with blockage ratio of 0.25, Reynolds numbers of 50 to 375 on a 241×81 uniform grid.

For the purposes of computations a parabolic profile is assumed at the inlet, no-slip conditions at the walls and at the obstacle, and the outlet boundary conditions imposed at the outlet (L/B= 24.0), are the convective boundary conditions as proposed by Orlanski[139]. The equations being solved are (5.36), (5.37) and (5.38), in which pseudo pressure ψ is used. The boundary conditions for ψ are $\psi = 0$ at the exit and homogeneous Neumann on the solid walls, obstacle and the inlet. The use of pseudo pressure simplifies the boundary conditions but at the cost of are not being able to compute C_L and C_D which require knowledge of the true pressure.

Figure 5.40 show the stream function plots of the flow field for Reynolds number of 60, 65, 70 and 75, respectively, all for a blockage ratio of 0.25. It may be observed that the flow is steady (and symmetrical) for Reynolds number of 60 while for Re=65 and above signs of unsteadiness and vortex shedding appear. Thus, in our calculations for the assumed blockage ratio the critical Reynolds number is around 65. Davis et al[132] numerically obtained a critical Reynolds number based on centerline velocity of about 100.0 which corresponds to Re=66.66 as per our definition of Reynolds number. This seems to validate our figure. Mukhopadhyay et al[93] obtains a critical Reynolds number of 85 using the EXTRA-FLAG scheme. However, that scheme was somewhat more diffusive than the OCV scheme and numerical diffusion causes as overprediction of the critical Reynolds number.

Figure 5.41 shows the time variation of v component of velocity on the centerline at a point 2B units behind the obstacle, for a typical case (Reynolds number of 375 and blockage ratio of 0.25) and its corresponding spectra is shown in Fig. 5.41 . It may be seen that the velocity variation is essentially periodic with a single dominant frequency. The Strouhal number computed from our results for Reynolds number of 150 is 0.2868. Davis et al[132] obtained a Strouhal number of 0.2625 numerically and 0.302 experimentally for a Reynolds number of 166.6 (Their definition of Reynolds number and Strouhal number are based centerline velocity U_o and are here converted
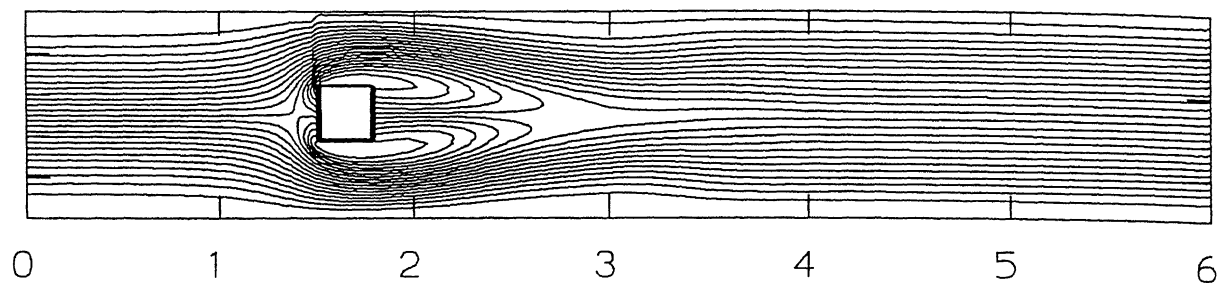
(i)
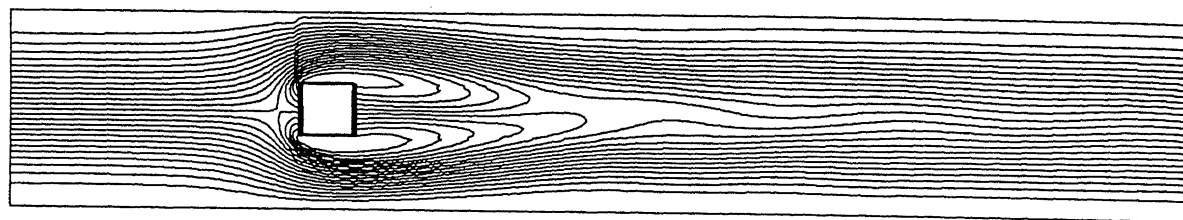
(ii)
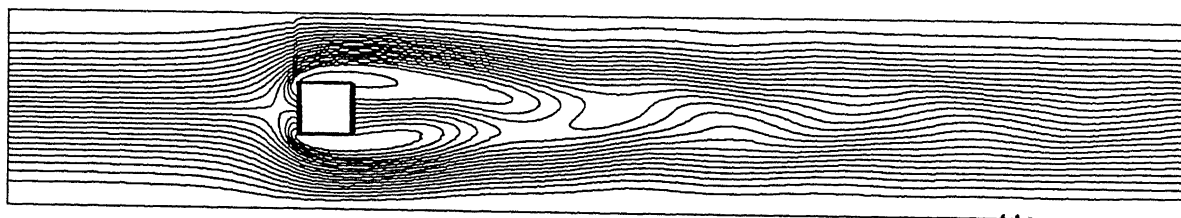
(iii)
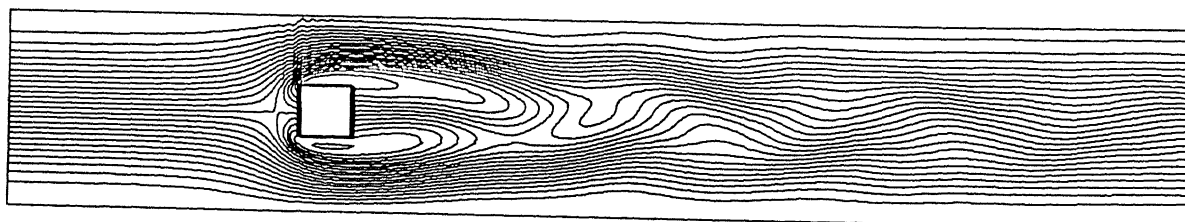
(iv)

Figure 5.40: Streamline plots of the flow field for (i) Re=60, (ii) Re=65, (iii) Re=70, (iv) Re=75

into our's by assuming $U_o = 1.5\ U_{av}$. Mukhopadhyay et. al.[136] obtains a somewhat lower Strouhal number of 0.2384 for the same case (Re=150 and blockage ratio=0.25). In our calculations we further obtain Strouhal number values of 0.2723 and 0.3168 for Reynolds number of 75 and 375, respectively. Figures 5.42(a)-(g) shows a velocity plot spanning an entire cycle of vortex shedding for Reynolds number 150 and blockage ratio of 0.25. The figures show qualitatively much the same behaviour as reported in the literature.
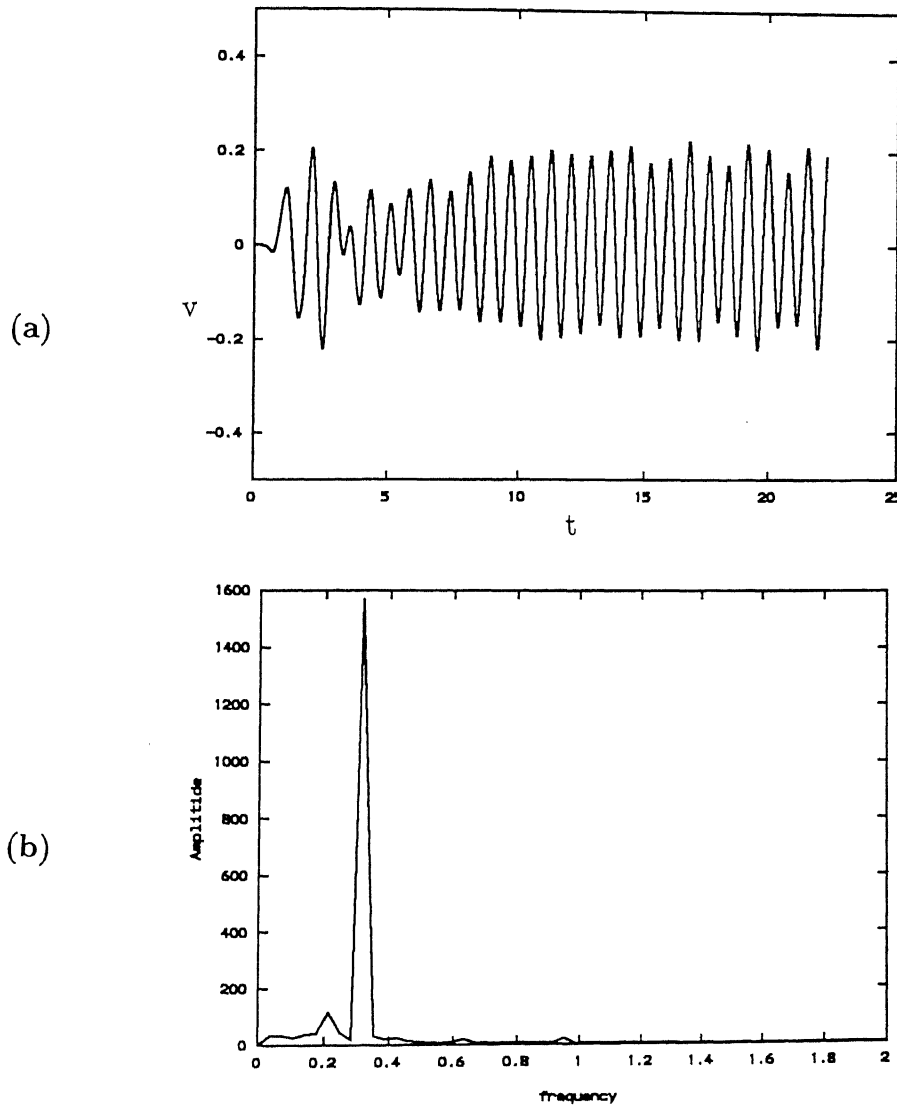

Figure 5.41: (a) Time variation of v component of velocity on the centerline at a point 2B units behind the obstacle and (b) its spectra for Reynolds number of 375 and blockage ratio=0.25
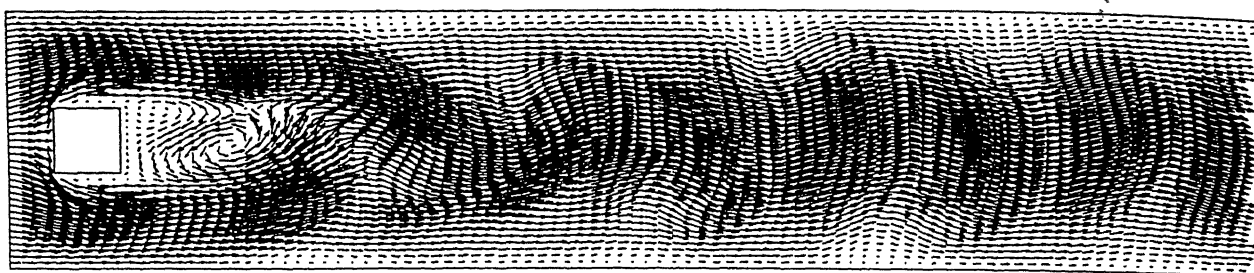
(a)
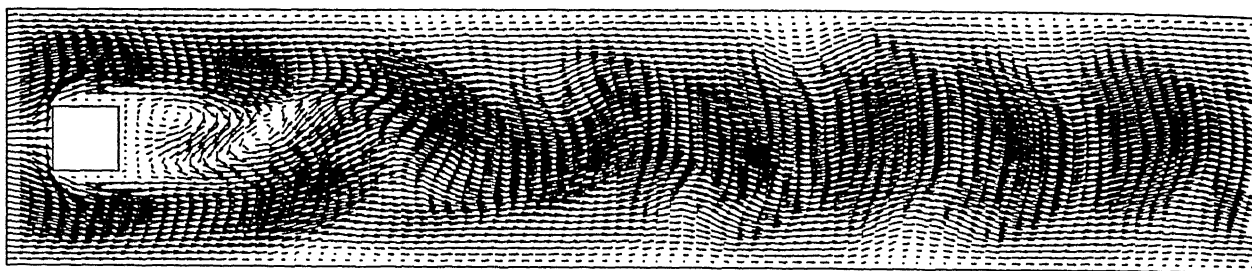
(b)
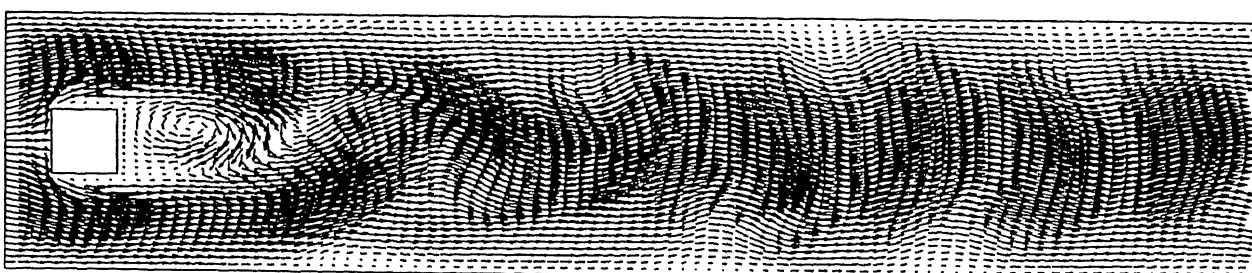
(c)
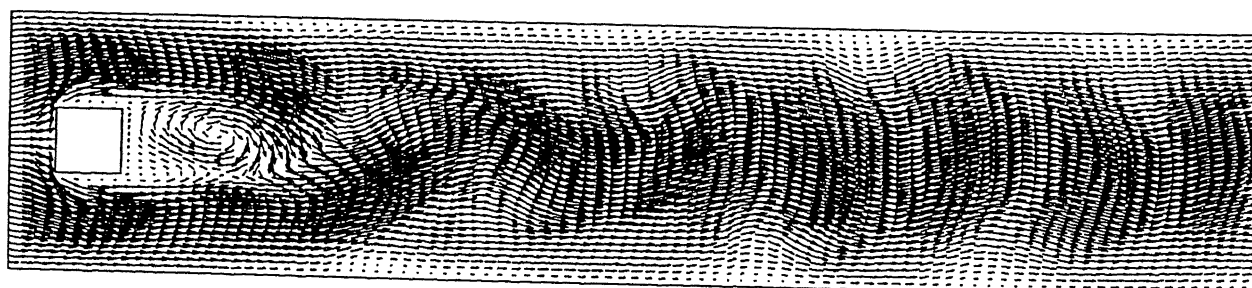
(d)

Figure 5.42: The velocity vector plot for Re=150 and Blokage-ratio=0.25
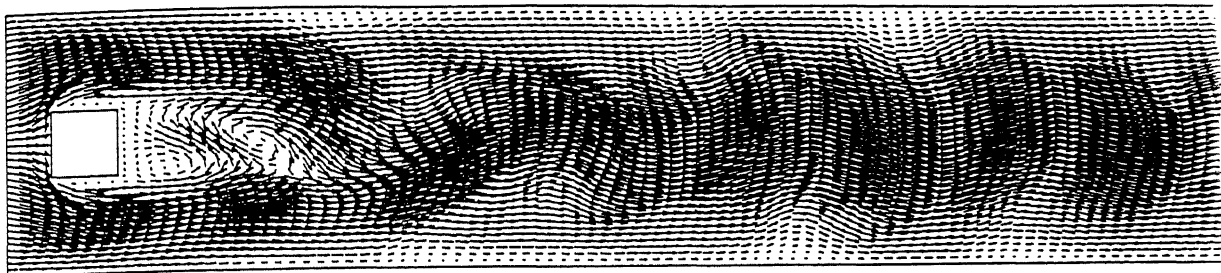
(e)
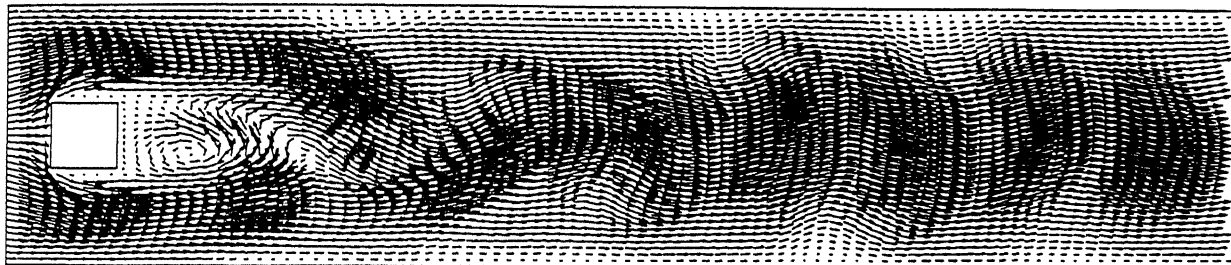
(f)
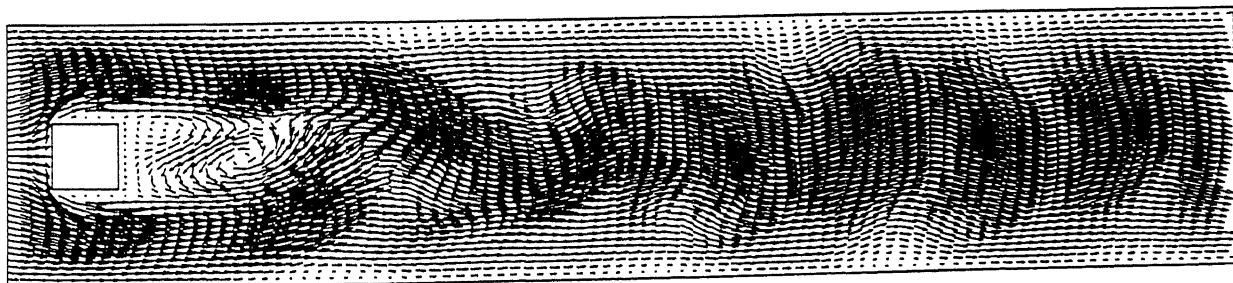
(g)

Figure 5.42[contd]. Velocity vector plot for test problem 6 (Re=150 and B/H=0.25)

5.6 Summary

An algorithm for computing steady and unsteady solutions to the two-dimensional incompressible Navier-Stokes equations on non-staggered grids in primitive variable formulation has been presented. This algorithm works well on both orthogonal and nonorthogonal grids. The salient features of this study have been summarized below.

- The physical domain is discretized into overlapping control volumes. Since control volumes are formed by take the grid points as the vertices of the cells, grid point information is directly used to compute geometrical parameters.

- Since the problem is solved in the physical domain, the governing partial differential equations are simpler than their counterpart in the generalized coordinate system.

- An iso-parametric description of the variables and geometry have been used. However the spirit of the method is that of finite volume rather than that of finite-element methods, as no assembly is used.

- The Cartesian components of velocities are used in the calculations and there are no complications associated with the use of covariant or contravariant components of velocities.

- An equal-order interpolation is used for the velocity components and pressure. The problem of checkerboard pressure distribution is avoided by using *momentum interpolation*.

- The proposed method is validated by computing the laminar flow in six test cases. In all cases, computed results exhibit good agreement with reference solutions.

- The applicability of the method to nonorthogonal grids is also demonstrated. The accuracy of the solution is not very sensitive to the moderate grid nonorthogonality.

- The algorithm has also been shown to be effective in obtaining unsteady solutions for the non-trivial problem of predicting vortex-shedding behind a square cylinder. The results obtained match well with those in literature.

References

[1] P.J. Roache, 'Computational Fluid Dynamics', Hermosa Publ., Albuquerque, New Mexixo (1972).

[2] D.B. Spalding, 'A novel finite-difference formulation for differential expressions involving both first and second derivatives', Int. J. Numer. Methods Eng., 4, 551-559 (1972).

[3] G.D. Raithby and K.E. Torrance, 'Upstream weighted schemes and their application to elliptic problems involving fluid flow', Comput. Fluids, 2, 191-206 (1974).

[4] S.V. Patankar, Numerical Heat Transfer and fluid flow, Hemisphere, NewYork, 1980.

[5] H.S. Price, R.S. Varga and J.E. Warren, 'Applications of oscillation matrices to diffusion-correction equations', J. Mathematics and Physics, 45, 301-311 (1966).

[6] B.P. Leonard, 'A stable and accurate convective modelling procedure based on quadratic upstream interpolation', Comput. Methods Appl. Mech. Eng., 19, 59-98 (1979).

[7] C.A.J. Fletcher, Computational techniques for fluid dynamics, Vol. II, Springer Verlag, New York, 1988.

[8] A.J. Baker, Finite element computational fluid mechanics, Hemisphere Publishing Corporation, New York, 1983.

[9] J.L. Steger and R.L. Sorenson, 'Automatic mesh point clustering near a boundary in grid generation with elliptic partial differential equations', J. Comput. Phys., 33, 405-410 (1979).

[10] R.L. Sorenson and J.L. Steger, 'Numerical generation of two dimensional grids by the use of poisson equations with grid control at boundaries', Numerical grid generation Techniques, NASA Conf. Pub., 2166 (1980).

[11] J.F. Thompson, J.U.A. Warsi and C.W. Mastin, 'Boundary fitted coordinate system for numerical solution of partial diffrential equations- A review', J. Comput. Phys., 47, 1-108 (1982).

[12] C.M. Rhie and W.L. Chow, 'Numerical study of the turbulent flow past an airfoil with trailing separation', AIAA J., 21, 1325-1332(1983).

[13] P.J. Roache, 'On artificial viscosity', J. Comput. Phys., 10, 169-184 (1972).

[14] G. de Vahl Davies and G.D. Mallinson, 'An evaluation of upwind and central difference approximations by study of recirculating flows', Comput. Fluids, 4, 29-43 (1976).

[15] G.D. Raithby, 'A critical evaluation of upstream differencing applied to problems involving fluid flow', Comput. Methods Appl. Mech. Eng., 9, 75-103 (1976).

[16] M.A. Leschziner,'Practical evaluation of three finite difference schemes for the computation of steady-state recirculating flows', Comput. Meth. Appl. Mech. Eng., 23, 293-312 (1980).

[17] G.D. Raithby, 'Skew upstream differencing schemes for problems involving fluid flow', Comput. Methods Appl. Mech. Eng., 9, 153-164 (1976).

[18] T. Han, J.A.C. Humphrey, and B.E. Launder, 'A comparison of hybrid and quadratic upstream differencing in high Reynolds number elliptic flows', Comput. Methods Appl. Mech. Eng., 29, 81-95 (1981).

[19] A. Pollard and L.W.A. Siu, 'The calculation of some laminar flows using various discretization schemes', Comput. Methods Appl. Mech. Eng., 35, 293-313 (1982).

[20] C.J. Freitas, R.L. Street, A.N. Findikakis and J.R. Koseff, 'Numerical simulation of three dimensional flow in a cavity', Int. J. Numer. Methods Fluids, 5, 561-575 (1985).

[21] W. Shyy, S.S. Tong, S.M. Correa, ' Numerical recirculating flow calculation using a body fitted coordinate system', Numer. Heat Transfer, 8, 99-113 (1985).

[22] S.P. Vanka, 'Second-order upwind differencing in a recirculating flow', AIAA J., 25, 1435-1441 (1987).

[23] B. Engquist and S. Osher, 'One-sided difference approximations for nonlinear conservation laws', Mathematics of Computaion, 36, 321-352 (1981).

[24] S.K. Godunov, 'A difference scheme for numerical computation of discontinuous solution of hydrodynamic equations', Math. Sbornik, 41, 271-306 (1959) [in Russian]. Translated US Joint Pub. Res. Service, JPRS 7226 (1969).

[25] A. Harten, 'High resolution schemes for hyperbolic conservations laws', J. Comput. Phys., 49, 357-393 (1983).

[26] B. Van Leer, Toward the ultimate conservative schemes. I. The quest for monotonicity, Lecture Notes in Physics, 18, 163-168 (1973). Springer Verlag, Berlin.

[27] J.P. Boris and D.L. Book, 'Flux-corrected transport I : SHASTA, a fluid transport algorithm that works', J. Comput. Phys., 11, 38-69 (1973).

[28] S.T. Zalesak, 'Fully multi-dimensional flux-corrected transport algorithms for fluids', J. Comput. Phys., 31, 335-362 (1979).

[29] C.W. Shu and S. Osher, 'Efficient implementation of essentially non-oscillatory shock-capturing schemes', J. Comput. Phys., 77, 439-471, (1988).

[30] H. Yang, 'An artificial compression method for ENO schemes: the slope modification method', J. Comput. Phys., 89, 125-160 (1990).

[31] J.C.T. Wang and G.F. Windhopf, 'A high-resolution TVD finite volume scheme for the Euler equations in conservative form', J. Comput. Phys., 84, 145-173 (1989).

[32] C. Hirsch, Numerical Computation of Internal and External Flows, Vol. 2, Wiley, New York, 1990.

[33] P.K. Sweby, 'High resolution schemes using flux limiters for hyperbolic conservation laws', SIAM J. Numer. Anal., 21, 995-1001 (1984).

[34] B.P. Leonard, Universal limiter for transient interpolation modelling of the advective transport equations : the ULTIMATE conservative difference scheme, NASA Technical Memorandum 100916, ICOMP-88-11 (September 1988).

[35] P.H. Gaskel and A.K.C. Lau, 'Curvature compensated convective transport : SMART, a new boundedness preserving transport algorithm', Int. J. Numer. Methods Fluids, 8, 617-647 (1988).

[36] B.P. Leonard, 'Simple high-accuracy resolution program for convective modelling of discontinuities', Int. J. Numer. Methods Fluids, 8, 1291-1318 (1988).

[37] B. Van Leer, 'Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection', J. Comput. Phys., 23, 276-299 (1977).

[38] M.S. Darwish and F.H. Moukalled, 'Normalized variable and space formulation methodology for high resolution schemes', Numer. Heat Transfer, Part B, 26, 79-96 (1994).

[39] J. Zhu and M.A. Leschziner, 'A local oscillation-damping algorithm for higher-order convection schemes', Comput. Methods Appl. Mech. Engg., 67, 355-366 (1988).

[40] A.K. Runchal, 'CONDIF : A modified central difference scheme for convective flows', Int. J. Numer. Methods Eng., 24, 1593-1608 (1987).

[41] F.H. Harlow and J.E. Welch, 'Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface', The Phys. Fluids, 8, 2182-2188 (1965).

[42] A.A. Amsdon and F.H. Harlow, The SMAC method : A numerical technique for calculating incompressible fluid flows, Los Alamos Scientific Laboratory, Los Alamos, NM, University of California, LA-4370, 1970.

[43] S. Abdallah, 'Numerical solutions for the pressure Poisson equation with Neumann boundary conditions using a non-staggered grid – I, J. Comput. Phys., 70, 182 (1987).

[44] L. Cheng and Steven Armfield, 'A simplified marker and cell method for unsteady flows on non-staggered grids', Int. J. Numer. Methods Fluids, 21, 15-34 (1995).

[45] M.F. Tome and S. Mckee, 'GENSMAC : A computational marker and cell method for free surface flows in general domains', J. Comput. Phys., 110, 171-186 (1994).

[46] A.J. Chorin, 'A numerical method for solving incompressible viscous flow problems', J. Comput. Phys., 2, 12-26 (1967).

[47] A.J. Chorin, 'Numerical solution of the Navier-Stokes equations', Math. Comput., 22, 745-762(1968).

[48] S.V. Patankar and D.B. Spalding, 'A calculation procedure for heat mass and momentum transfer in three dimensional parabolic flows', Int. J. Heat and Mass Trans., 15, 1787-1806 (1972).

[49] S.D. Connell and P. Slow, 'The pressure-correction method', Comput. Fluids, 14, 1-10 (1986).

[50] X. Wen and D.B. Ingham, 'A new method for accelerating the rate of convergence of the SIMPLE-like algorithm', Int. J. Numer. Methods Fluids', 17, 385-400(1993).

[51] J.P. VanDoormal and G.D. Raithby, ' Enhancement of the SIMPLE method for predicting incompressible fluid flows', Numer. Heat Transfer, 7, 147-163 (1984).

[52] R.I. Issa, 'Solution of implicitly discretized fluid flow equation by operator-splitting', J. Comput. Phys., 62, 40-65(1985).

[53] D.S. Jang, R. Jetli and S. Acharya, 'Comparison of PISO, SIMPLER and SIMPLEC algorithms for the treatment of the pressure velocity coupling in steady flow problems', Numer. Heat Transfer, 10, 209-228 (1986).

[54] M. Peric, R. Kessler and G. Scheuerer, 'Comparison of finite volume numerical methods with staggered and colocated grids', Comput. Fluids, 16, 389-403 (1988).

[55] T.F. Miller and F.W. Schmidt, 'Use of a pressure-weighted interpolation method for the incompressible Navier Stokes equations on a non-staggered grid system', Numer. Heat. Transfer, 14, 213-233 (1988).

[56] M.H. Kobayashi and J.C.F. Pereira, 'Calculation of incompressible flows on a nonstaggered, non-orthogonal grid', Numer. Heat Transfer, 19, 243 (1991).

[57] S. Majumdar, 'Role of under relaxation in momentum interpolation for calculation of flow with non-staggered grids', Numer. Heat Transfer, 13, 125-132 (1988).

[58] A.W. Date, 'Solution of Navier-Stokes equations on non-staggered grid', Int. J. Heat Mass Transfer, 36, 1913-1922 (1993).

[59] A.W. Date, 'Complete pressure correction algorithm for solution of incompressible Navier-Stokes equations on a nonstaggered grid', Numer. Heat Transfer, Part B, 29, 441-458 (1996).

[60] H. Aksoy and C.J. Chen, 'Numerical solution of Navier-Stokes equations with nonstaggered grids using finite analytic method', Numer. Heat Transfer, Part-B, 21, 287-306 (1992).

[61] K.C. Karki and S.V. Patankar, 'Calculation procedure for viscous incompressible flows in complex geometries', Numer. Heat Transfer, 14, 295-307 (1988).

[62] T.K. Hung and T.D. Brown, 'An implicit finite difference method for solving the Navier Stokes equations using orthogonal curvilinear coordinates', J. Comput. Phys., 23, 343-363 (1977).

[63] S.B. Pope, 'The calculation of turbulent recirculating flows in general orthogonal coordinates', J. Comput. Phys., 26, 197-217 (1978).

[64] C.W. Rapley, 'Turbulent flow in duct with cusped corners', Int. J. Numer. Methods Fluids, 5, 155-167 (1985).

[65] A.K. Rastogi, 'Hydrodynamics in tubes perturbed by curvilinear obstructions', ASME J. Fluid Eng., 106, 262-269 (1984).

[66] G.D. Raithby, P.F. Galpin and J.P. VanDoormal, 'Prediction of heat and fluid flow in complex geometries using general orthogonal coordinates', Numer. Heat Transfer, 9, 125-142 (1986).

[67] C.R. Maliska and G.D. Raithby, 'A method for computing three dimensional flows using non-orthogonal boundary fitted coordinates', Int. J. Numer. Methods Fluids, 4, 519-537 (1984).

[68] M. Vinokur, 'Conservative equations of gas dynamics in curvilinear coordinate systems', J. Comput. Phys., 14, 105-125 (1974).

[69] T. Gal-Chen and R.C.J. Somerville, 'Numerical solution of the Navier Stokes equations with topography', J. Comput. Phys., 17, 276-310 (1975).

[70] M. Faghri, E.M. Sparrow and A.T. Prata, 'Finite difference solutions of convection diffusion problems in irregular domains using a nonorthogonal coordinate transformation', Numer. Heat Transfer, 7, 183-209 (1984).

[71] I. Demirdzic, A.D. Gosman, R.I. Issa and M. Peric, 'A calculation procedure for turbulent flow in complex geometries', Comput. Fluids, 15, 251-273 (1987).

[72] M.C. Melaaen, 'Calculation of fluid flows with staggered and nostaggered curvilinear nonorthogonal grids - Theory', Numer. Heat Transfer A, 21, 1-19 (1992).

[73] T. Ikohagi and B.R. Shin, 'Finite-difference schemes for steady incompressible Navier-Stokes equations in general curvilinear coordinates', Comput. Fluids, 19, 479-488 (1991).

[74] S.W. Kim and T.J. Benson, 'Comparison of the SMAC, PISO and iterative time-advancing schemes for unsteady flows', Comput Fluids, 21, 435-454 (1992).

[75] T. Ikohagi, B.R. Shin and H. Daiguji, 'Application of an implicit time-marching scheme to a 3-dimensional incompressible flow problems in curvilinear coordinate systems', Comput. Fluids, 21, 163-175 (1992).

[76] C.F. Hsu, A curvilinear coordinate method for momentum heat and mass transfer in domains of irregular geometry, PhD Thesis, Univ. of Minnesota, 1981.

[77] M. Reggio and R. Camarero, 'Numerical solution procedure for viscous incompressible flows', Numer. Heat Transfer, 10, 131-146 (1986).

[78] S.W. Armfield, 'Finite difference solution of the Navier-Stokes equations on staggered and nonstaggered grids', Comput. Fluids, 20, 1-17(1991).

[79] D. Lee and J.J. Chiu, 'Covariant velocity-based calculation procedure with non-staggered grids for computation of pulsatile flows', Numer. Heat Transfer B, 21, 269-286 (1992).

[80] M. Peric, 'A finite volume method for prediction of three dimensional fluid flow in complex ducts', PhD Thesis, Univ. of London, 1985.

[81] W. Rodi, S. Majumdar and B. Schonung, 'Finite volume methods for two-dimensional incompressible flows with complex boundaries', Comput. Methods Appl. Mech. Eng., 75, 369-392 (1989).

[82] S. Majumdar, W. Rodi and J. Zhu, 'Three-dimensional finite-volume method for incompressible flow with complex boundaries', ASME J. Fluids Eng., 114, 496-503 (1992).

[83] I. Demirdzic, Z. Lilek and M. Peric, 'A collocated finite-volume method for predicting flows at all speeds', Int. J. Numer. Methods Fluids, 16, 1029-1050 (1993).

[84] B.R. Baliga and S.V. Patankar, 'A control volume finite element method for two dimensional fluid flow and heat transfer', Numer. Heat. Transfer, 6, 245-261 (1983).

[85] B.R. Baliga, T.T. Pham and S.V. Patankar, 'Solution of some two dimensional incompressible fluid flow and heat transfer problems, using a control volume finite-element method', Numer. Heat Transfer, 6, 263-282 (1983).

[86] C. Prakash and S.V. Patankar, 'A control volume-based finite-element method for solving the Navier-Stokes equations using equal-order velocity-pressure interpolation', 8, 259-280 (1985).

[87] C. Prakash, 'An improved control volume finite element method for heat and mass transfer and for fluid flow using equal order velocity-pressure interpolation', Numer. Heat. Transfer, 9, 253-276 (1986).

[88] B. LeDain-Muir and B.R. Baliga, 'Solution of three-dimensional convection-diffusion problems using tetrahedral elements and flow-oriented upwind interpolation functions', Numer Heat Transfer, 9, 253-276 (1986).

[89] N.A. Hookey and B.R. Baliga, 'Evaluation and enhancement of some control-volume finite-volume methods : Part I. Convection-diffusion problems', Numer. Heat Transfer, 14, 255-272 (1988).

[90] G.E. Schneider and M.J. Raw, 'Control volume finite-element method for the heat transfer and fluid flow using collocated variables – 1. Computational procedure', Numer. Heat Transfer, 11, 363-390 (1987).

[91] H.J. Saabas and B.R. Baliga, 'Co-located equal-order control-volume finite element method for multidimensional, incompressible, fluid flow– Part 1. Formulation, Numer. heat Transfer, 26, 381-407 (1994).

[92] C. Masson, H.J. Saabas and B.R. Baliga, 'Co-located equal-order control-volume method for two-dimensional axisymmetric incompressible fluid flow', Int. J. Numer. Methods Fluids, 18, 1-26 (1994).

[93] A. Mukhopadhyay, T. Sundararajan and G. Biswas,'An explicit transient algorithm for predicting incopmressible flows in arbitrary geometry, Int. J. Numer. Methods Fluids, 17, 975-993 (1993).

[94] J.P. Jessee and W.A. Fiveland, 'A cell vertex algorithm for the incompressible Navier-Stokes equations on non-orthogonal grids', Int. J. Numer. Methods Fluids, 23, 271-293 (1996).

[95] Atul Kumar Verma and V. Eswaran, 'Overlapping control volume approach for convection-diffusion equation', Int. J. Numer. Methods Fluids, 23, 865-882 (1996).

[96] Atul Kumar Verma and V. Eswaran, 'A bounded convection scheme for the overlapping control volume approach', Int. J. Numer. Methods Fluids (1997) [in print].

[97] Atul Kumar Verma and V. Eswaran, 'The OCV method for incompressible fluid flow calculations on nonstaggered grids' [Accepted for 14th NHMT and 3rd ISHMT/ASME conference to be held at IIT Kanpur, India].

[98] Atul Kumar Verma, S.M. Bhallamudi and V. Eswaran, 'Overlapping control volume method for solute transport' [communicated to International Journal for Numerical MEthods in Engineering].

[99] Atul Kumar Verma and V. Eswaran, 'A solution method of Navier-Stokes equations on nonstaggered grids' [communicated to International Journal for Numerical Methods in Fluids].

[100] C. Hirsch, Numerical Computation of Internal and External Flows, Vol. I, Wiley, New York, 1988.

[101] A.K. Runchal, 'Convergence and accuracy of three finite differences schemes for a two-dimensional conduction and convection problem', Int. J. Numer. Methods Eng., 4, 541-550 (1972).

[102] Y.H. Hwang, 'Arbitrary domain velocity analyses for the incompressible Navier-Stokes equations ', J. comput. phys., 110, 134-149 (1994).

[103] P. K. Khosla and S. G. Rubin, 'A diagonally dominant second-order accurate implicit scheme', Comput. Fluids, 2, 207-218 (1974).

[104] R.M. Smith and A.G. Hutton, 'The numerical treatment of advection : a performance comparison of current methods ', Numer Heat Transfer, 5, 439-461 (1982).

[105] Panos Tamamidis and D.N. Assanis, 'Evaluation of various high-order-accuracy schemes with and without flux limiters', Int. J. Numer. Methods Fluids, 16, 931-948 (1993).

[106] P.L. Roe, 'Some contributions to the modeling of discontinuous flows', Proc. AMS/SIAM Seminar, San Diego, 1983.

[107] P.S. Huyakorn and G.F. Pinder, Computational Methods in Subsurface Flow, Academic, San Diego, Calif. (1983).

[108] Van Genuchten, On the accuracy and efficiency of several numerical schemes for solving the convective-dispersive equation, in Finite elements in Water Resource (Eds. W.G. Gray, G.F. Pinder, and I.A. Brebbia) (1977).

[109] G.F. Pinder and Shapiro, 'A new collocation method for the solution of the convection dominated transport equation', Water Resource Research, 15, 1177-1182 (1979).

[110] J.C. Heinrich, P.S. Huyakorn, O.C. Zienkiewicz and A.R. Mitchell, 'An upwind finite element for two-dimensional convective transport equation', Int. J. Numer. Methods in Engg., 11, 131-143 (1977).

[111] N.Z. Sun and W.W.G. Yeh, 'A proposed upstream weight numerical method for simulating pollutant transport in ground water', Water Resource Research, 19, 1489-1500 (1993).

[112] C. Wang, N.Z. Wang and W.W.G. Yeh, 'An upstream weight multiple cell balance finite-element method for solving three-dimensional convection-dispersion equations', Water Resource Research, 22, 1575-1589 (1986).

[113] G.T. Yeh, 'An orthogonal upstream finite-element approach to modelling aquifer contaminant transport', Water Resource Research, 22, 952-964 (1986).

[114] J.J. Westerink and D. Shea, 'Consistent higher degree Petro-Galerkin methods for the solution of transient-diffusion equation, Int. J. Numer. Methods Engg., 28, 1077-1100 (1986).

[115] F.X. Yu and V.P. Singh, 'Improved finite element method for solute transport', J. Hydraulic Engg., ASCE, 121, 145-158 (1995).

[116] G.T. Yeh, 'A Lagrangian-Eulerian method with zoomable hidden fine-mesh approach for solving advection-dispersion equations', Water Resources Research, 1133-1144 (1990).

[117] G.T. Yeh and J.R. Chang, 'An exact peak capturing and oscillation-free scheme to solve advection-dispersion transport equations', Water Resource Research, 28, 2937-2951 (1992).

[118] Y. Ijiri and K. Karasaki, 'A Lagrangian-Eulerian finite element method with adaptive gridding for advection-dispersion problems', Computational Methods in Water Resources, X, 291-298 (1994).

[119] R. Peyret and T.D. Taylor, Computational Methods for Fluid Flow, Springer Verlag, New York, 1983.

[120] M. Putti, W.W.G. Yeh and W.A. Mulder, 'A triangular finite volume approach with high resolution upwind terms for the solution of ground water transport equation', Water Resource Research, 26, 2865-2880 (1990).

[121] R.A. Cox and J. Nishikawa, 'A new Total Variation Diminishing scheme for the solution of advective-dominant solute transport, Water Resource Research, 27, 2645-2654 (1991).

[122] R. A.Freeze and J. A. Cherry, Groundwater, Prentice-Hall, Englewood Cliffs, N.J. (1979).

[123] Vedat Batu, 'A generalized two-dimensional analytical solution for hydrodynamic dispersion in bounded media with the first-type boundary condition at the source', Water Resource Research, 25, 1125-1132 (1989).

[124] Vedat Batu, 'A generalized two-dimensional analytical solute transport model in bounded media for flux-type finite multiple sources', Water Resources Research, 29, 2881-2892 (1993).

[125] I. Javandel, C. Doughty and C.F. Tsang, Groundwater transport, Handbook of Mathematical Models, American Geophysical Union, 14-19 (1984).

[126] U. Ghia, K.N. Ghia and C.T. Shin, 'High resolutions for incompressible flow using the Navier-Stokes equations and a multigrid method', J. Comput. Phys., 48, 387-411 (1982).

[127] M.P. Reddy, L.G. Reifschneider, J.N. Reddy and H.U. Akay, 'Accuracy and convergence of element-by-element iterative solvers for incompressible fluid flows using penalty finite element model', Int. J. Numer. Methods. Fluids, 17, 1019-1033 (1993).

[128] B.F. Armaly, F. Durst, J.C.F. Pereira and B. Schonung, 'Experimental and theoretical investigation of backward-facing step flow', J. Fluid Mech., 172, 473-496 (1983).

[129] J. Kim and P. Moin, 'Application of a fractional-step method to incompressible Navier-Stokes equations', J. Comput. Phys., 59, 308-323 (1985).

[130] M.C. Thompson and J.H. Ferziger, 'An adoptive multigrid technique for the incompressible Navier-Stokes equations', J. Comput. Phys., 82, 94-121 (1989).

[131] D.K. Gartling, 'A test problem for outflow boundary conditions – Flow over a backward-facing step', Int. J. Numer. Methods Fluids, 11, 953-967 (1990).

[132] R.W. Davis, E.F. Moore, and L.P. Purtell, 'A numerical-experimental study of confined flow around rectangular cylinders', Phys. Fluids, 27, 46-59 (1984).

[133] A. Okajima, ' Strouhal numbers of rectangular cylinders', J. Fluid Mech, 123, 379-398 (1982).

[134] A. Okajima, 'Numerical simulation of flow around rectangular cylinders', J. Wind Eng. and Ind. Aerodynamics, 33, 171-180 (1990).

[135] T. Tamura, 'Numerical study of aerodynamic behaviour of a square cylinder', J. Wind Eng. and Ind. Aerodynamics, 33, 161-170 (1990).

[136] A. Mukhopadhyay, G. Biswas and T. Sundarajan, 'Numerical investigation of confined wakes behind a square cylinder in a channel', Int J. Numer. Methods Fluids, 14, 1473-1484 (1992).

[137] K.M. Kelkar and S.V. Patankar, 'Numerical prediction of vortex shedding behind a square cylinder', Int. J. Numer. Methods Fluids, 14, 327-341 (1992).

[138] G. Li and J.A.C. Humphrey, 'Numerical modelling of confined flow past a cylinder of square cross-section at various orientations', Int. J. Numer. Methods Fluids, 20, 1215-1236 (1995).

[139] I. Orlanski, 'A simple boundary condition for unbounded hyperbolic flows', J. Comput. Phys., 21, 251-269 (1976).

Appendix A

Diffusion Term :

For the face 1, at the mid-point (i.e. $\xi = 0, \eta = -1$) of the control surface as shown in Figures 2.2 and 2.3, the diffusion term is shown below :

$$\text{DIFF\_1}(k) = \Gamma(\frac{\partial N_k}{\partial x} \mid_{(0,-1)} \Delta y^{(1)} - \frac{\partial N_k}{\partial y} \mid_{(0,-1)} \Delta x^{(1)})$$

where k=1-5 are the local node numbers in the counter clockwise sense as shown in Figure 2.3. The other terms on the right hand side of the above expression have already been defined in section 2. Similarly for the control-volume faces 2, 3, and 4 the diffusion coefficients are, respectively, as follows :

$$\text{DIFF\_2}(k) = \Gamma(\frac{\partial N_k}{\partial x} \mid_{(1,0)} \Delta y^{(2)} - \frac{\partial N_k}{\partial y} \mid_{(1,0)} \Delta x^{(2)})$$

$$\text{DIFF\_3}(k) = \Gamma(\frac{\partial N_k}{\partial x} \mid_{(0,1)} \Delta y^{(3)} - \frac{\partial N_k}{\partial y} \mid_{(0,1)} \Delta x^{(3)})$$

$$\text{DIFF\_4}(k) = \Gamma(\frac{\partial N_k}{\partial x} \mid_{(-1,0)} \Delta y^{(4)} - \frac{\partial N_k}{\partial y} \mid_{(-1,0)} \Delta x^{(4)})$$

where k=1-5. If we define local nodes 1, 2, 3, and 4 as the west, south, east, and north neighbours of the node P as shown in Figure 2.2. The final expression for the diffusion coefficients for a control-volume can be expressed as

$D_N = \text{DIFF\_1}(4) + \text{DIFF\_2}(4) + \text{DIFF\_3}(4) + \text{DIFF\_4}(4)$

$D_S = \text{DIFF\_1}(2) + \text{DIFF\_2}(2) + \text{DIFF\_3}(2) + \text{DIFF\_4}(2)$

$D_E = \text{DIFF\_1}(3) + \text{DIFF\_2}(3) + \text{DIFF\_3}(3) + \text{DIFF\_4}(3)$

$D_W = \text{DIFF\_1}(1) + \text{DIFF\_2}(1) + \text{DIFF\_3}(1) + \text{DIFF\_4}(1)$

$D_P = \text{DIFF\_1}(5) + \text{DIFF\_2}(5) + \text{DIFF\_3}(5) + \text{DIFF\_4}(5)$

Convection Term

For face 1, again at the mid-point, the convection term is approximated as

$$\begin{aligned} \text{CONV\_1} &= (\phi^{(1)} F^{(1)}) \,|_{mid} \\ &= (N_k \phi_k)^{(1)} F^{(1)} \,|_{mid}, \quad k = 1 - 5 \end{aligned}$$

Here, mid-point is ($\xi=0$, $\eta=-1$) for the positive value of $F^{(1)}$ and ($\xi=0$, $\eta=1$) for the negative $F^{(1)}$.Combining both the possibilities in a single expression, we get

$$\begin{aligned} \text{CONV\_1} &= \max(F^{(1)}, 0)[C\_1(1)\phi_{i-1,j} + C\_1(2)\phi_{i,j-1} + C\_1(3)\phi_{i+1,j} + C\_1(4) \\ &\quad \phi_{i,j+1} + C\_1(5)\phi_{i,j}] - \max(-F^{(1)}, 0)[C\_11(1)\phi_{i-2,j-1} + C\_11(2) \\ &\quad \phi_{i-1,j-2} + C\_11(3)\phi_{i,j-1} + C\_11(4)\phi_{i-1,j} + C\_11(5)\phi_{i-1,j-1}] \end{aligned}$$

Similar expressions for the other surfaces (CONV_2, CONV_3, and CONV_4) of a control-volume can be obtained. Now rearranging the terms and writing expressions for each node of a control-volume, we get

$$\begin{aligned} C_W &= [C\_1(1)\max(F^{(1)}, 0) + C\_2(1)\max(F^{(2)}, 0) + C\_3(1)\max(F^{(3)}, 0) \\ &\quad + C\_4(1)\max(F^{(4)}, 0)] - [C\_11(4)\max(-F^{(1)}, 0) + C\_44(2)\max(-F^{(4)}, 0)] \end{aligned}$$

$$\begin{aligned} C_E &= [C\_1(3)\max(F^{(1)}, 0) + C\_2(3)\max(F^{(2)}, 0) + C\_3(3)\max(F^{(3)}, 0) \\ &\quad + C\_4(3)\max(F^{(4)}, 0)] - [C\_22(4)\max(-F^{(2)}, 0) + C\_33(2)\max(-F^{(3)}, 0)] \end{aligned}$$

$$\begin{aligned} C_N &= [C\_1(4)\max(F^{(1)}, 0) + C\_2(4)\max(F^{(2)}, 0) + C\_3(4)\max(F^{(3)}, 0) \\ &\quad + C\_4(4)\max(F^{(4)}, 0)] - [C\_33(1)\max(-F^{(3)}, 0) + C\_44(3)\max(-F^{(4)}, 0)] \end{aligned}$$

$$\begin{aligned} C_S &= [C\_1(2)\max(F^{(1)}, 0) + C\_2(2)\max(F^{(2)}, 0) + C\_3(2)\max(F^{(3)}, 0) \\ &\quad + C\_4(2)\max(F^{(4)}, 0)] - [C\_11(3)\max(-F^{(1)}, 0) + C\_22(1)\max(-F^{(2)}, 0)] \end{aligned}$$

$$\begin{aligned} C_P &= [C\_1(5)\max(F^{(1)}, 0) + C\_2(5)\max(F^{(2)}, 0) + C\_3(5)\max(F^{(3)}, 0) \\ &\quad + C\_4(5)\max(F^{(4)}, 0)] \end{aligned}$$

where the subscripts N, S, E, and W denotes the neighbouring nodes as defined earlier
and the remaining terms of CONV_1, CONV_2, CONV_3, and CONV_4 can be included
in the term b of equation (2.19) Finally, the coefficients can be represented as

$$
\begin{aligned}
a_P &= C_P - D_P + \max(C_N, 0) + \max(C_S, 0) + \max(C_E, 0) + \max(C_W, 0) \\
a_N &= D_N + \max(-C_N, 0) \\
a_S &= D_S + \max(-C_S, 0) \\
a_E &= D_E + \max(-C_E, 0) \\
a_W &= D_W + \max(-C_W, 0) \\
b &= SS - \max(C_N, 0)(\phi_{i,j+1} - \phi_{i,j}) - max(C_S, 0)(\phi_{i,j-1} - \phi_{i,j}) \\
&\quad - \max(C_E, 0)(\phi_{i+1,j} - \phi_{i,j}) - \max(C_W, 0)(\phi_{i-1,j} - \phi_{i,j})
\end{aligned}
$$

where SS consists of source terms as well as any other terms which can not be included
in the other coefficients defined above.

Appendix B

Diffusion Term :

For the face 1 (i.e. k=1), at the mid-point (i.e. $\xi = 0, \eta = -1$) of the control surface as shown in Figures 2.2 and 2.3, the diffusion coefficients are shown below :

$$
\begin{aligned}
\text{DIFF\_1}(kk) &= \left(D_{xx}^{(1)} \frac{\partial N_{kk}}{\partial x} \Big|_{(0,-1)} \Delta z^{(1)} + D_{xz}^{(1)} \frac{\partial N_{kk}}{\partial z} \Big|_{(0,-1)} \Delta z^{(1)} \right. \\
&\quad \left. - D_{zx}^{(1)} \frac{\partial N_{kk}}{\partial x} \Big|_{(0,-1)} \Delta x^{(1)} - D_{zz}^{(1)} \frac{\partial N_{kk}}{\partial z} \Big|_{(0,-1)} \Delta x^{(1)} \right)
\end{aligned}
$$

where kk=1-5 are the local node numbers in the counter clockwise sense as shown in Figure 2.3. The other terms on the right hand side of the above expression have already been defined in section 2. Similarly for the control-volume faces 2, 3, and 4 the diffusion coefficients are, respectively, as follows :

$$
\begin{aligned}
\text{DIFF\_2}(kk) &= \left(D_{xx}^{(2)} \frac{\partial N_{kk}}{\partial x} \Big|_{(1,0)} \Delta z^{(2)} + D_{xz}^{(2)} \frac{\partial N_{kk}}{\partial z} \Big|_{(1,0)} \Delta z^{(2)} \right. \\
&\quad \left. - D_{zx}^{(2)} \frac{\partial N_{kk}}{\partial x} \Big|_{(1,0)} \Delta x^{(2)} - D_{zz}^{(2)} \frac{\partial N_{kk}}{\partial z} \Big|_{(1,0)} \Delta x^{(2)} \right)
\end{aligned}
$$

$$
\begin{aligned}
\text{DIFF\_3}(kk) &= \left(D_{xx}^{(3)} \frac{\partial N_{kk}}{\partial x} \Big|_{(0,1)} \Delta z^{(3)} + D_{xz}^{(3)} \frac{\partial N_{kk}}{\partial z} \Big|_{(0,1)} \Delta z^{(3)} \right. \\
&\quad \left. - D_{zx}^{(3)} \frac{\partial N_{kk}}{\partial x} \Big|_{(0,1)} \Delta x^{(3)} - D_{zz}^{(3)} \frac{\partial N_{kk}}{\partial z} \Big|_{(0,1)} \Delta x^{(3)} \right)
\end{aligned}
$$

$$\text{DIFF\_4}(kk) = \left(D_{xx}^{(4)} \frac{\partial N_{kk}}{\partial x} \Big|_{(-1,0)} \Delta z^{(4)} + D_{xz}^{(4)} \frac{\partial N_{kk}}{\partial z} \Big|_{(-1,0)} \Delta z^{(4)} \right.$$

$$\left. - D_{zx}^{(4)} \frac{\partial N_{kk}}{\partial x} \Big|_{(-1,0)} \Delta x^{(4)} - D_{zz}^{(4)} \frac{\partial N_{kk}}{\partial z} \Big|_{(-1,0)} \Delta x^{(4)} \right)$$

where kk=1- 5. If we define local node 1, 2, 3, and 4 as the west, south, east, and north neighbours of the node p. The final expression for the diffusion coefficients for a control-volume can be expressed as

$D_W = \text{DIFF\_1}(1) + \text{DIFF\_2}(1) + \text{DIFF\_3}(1) + \text{DIFF\_4}(1)$

$D_S = \text{DIFF\_1}(2) + \text{DIFF\_2}(2) + \text{DIFF\_3}(2) + \text{DIFF\_4}(2)$

$D_E = \text{DIFF\_1}(3) + \text{DIFF\_2}(3) + \text{DIFF\_3}(3) + \text{DIFF\_4}(3)$

$D_N = \text{DIFF\_1}(4) + \text{DIFF\_2}(4) + \text{DIFF\_3}(4) + \text{DIFF\_4}(4)$

$D_P = \text{DIFF\_1}(5) + \text{DIFF\_2}(5) + \text{DIFF\_3}(5) + \text{DIFF\_4}(5)$

Convection Term

For face 1, again at the mid-point, the convection term is approximated as

$$\text{CONV\_1} = (C^{(1)} F^{(1)}) \big|_{mid}$$

$$= (N_k C_k)^{(1)} F^{(1)} \big|_{mid}, \quad k = 1 - 5$$

Here, mid-point is (ξ=0, η=-1) for the positive value of $F^{(1)}$ and (ξ=0, η=1) for the negative $F^{(1)}$. Combining both the possibilities in a single expression, we get

$$\text{CONV\_1} = \max(F^{(1)}, 0)[C\_1(1)C_{i-1,j} + C\_1(2)C_{i,j-1} + C\_1(3)C_{i+1,j} + C\_1(4)$$

$$C_{i,j+1} + C\_1(5)C_{i,j}] - \max(-F^{(1)}, 0)[C\_11(1)C_{i-2,j-1} + C\_11(2)$$

$$C_{i-1,j-2} + C\_11(3)C_{i,j-1} + C\_11(4)C_{i-1,j} + C\_11(5)C_{i-1,j-1}]$$

Similar expressions for the other surfaces (CONV_2, CONV_3, and CONV_4) of a control-volume can be obtained. Now rearranging the terms and writing expressions for each node of a control-volume, we get

$$C_W = [C\_1(1)\max(F^{(1)}, 0) + C\_2(1)\max(F^{(2)}, 0) + C\_3(1)\max(F^{(3)}, 0)$$

$$+ C\_4(1)\max(F^{(4)}, 0)] - [C\_11(4) \max(-F^{(1)}, 0) + C\_44(2)\max(-F^{(4)}, 0)]$$

$$C_E = [C\_1(3)\max(F^{(1)},0) + C\_2(3)\max(F^{(2)},0) + C\_3(3)\max(F^{(3)},0)$$
$$+ \ C\_4(3)\max(F^{(4)},0)] - [C\_22(4)\max(-F^{(2)},0) + C\_33(2)\max(-F^{(3)},0)]$$

$$C_N = [C\_1(4)\ \max(F^{(1)},0) + C\_2(4)\ \max(F^{(2)},0) + C\_3(4)\max(F^{(3)},0)$$
$$+ \ C\_4(4)\max(F^{(4)},0)] - [C\_33(1)\max(-F^{(3)},0) + C\_44(3)\max(-F^{(4)},0)]$$

$$C_S = [C\_1(2)\max(F^{(1)},0) + C\_2(2)\max(F^{(2)},0) + C\_3(2)\max(F^{(3)},0)$$
$$+ \ C\_4(2)\max(F^{(4)},0)] - [C\_11(3)\max(-F^{(1)},0) + C\_22(1)\max(-F^{(2)},0)]$$

$$C_P = [C\_1(5)\max(F^{(1)},0) + C\_2(5)\max(F^{(2)},0) + C\_3(5)\max(F^{(3)},0)$$
$$+ \ C\_4(5)\max(F^{(4)},0)]$$

where the subscripts N, S, E, and W denotes the neighbouring nodes as defined earlier and the remaining terms of CONV_1, CONV_2, CONV_3, and CONV_4 can be included in the term b of equation (4.12) Finally, the coefficients can be represented as

$$a_P = \theta\,[C_P - D_P + \max(C_N,0) + \max(C_S,0) + \max(C_E,0) + \max(C_W,0)]$$
$$+ \ \frac{R_d\ A_s}{\Delta t} + \theta\ \nu\ R_d\ A_s$$
$$a_N = \theta\,[D_N + \max(-C_N,0)]$$
$$a_S = \theta\,[D_S + \max(-C_S,0)]$$
$$a_E = \theta\,[D_E + \max(-C_E,0)]$$
$$a_W = \theta\,[D_W + \max(-C_W,0)]$$
$$b = SS + \frac{R_d\ A_s}{\Delta t}C_{i,j} - \theta\,[\max(C_N,0)(C_{i,j+1} - C_{i,j}) + max(C_S,0)(C_{i,j-1} - C_{i,j})$$
$$+\max(C_E,0)(C_{i+1,j} - C_{i,j}) + \max(C_W,0)(C_{i-1,j} - C_{i,j})]^n$$
$$+(1-\theta)[DIFF + CONV + DEC]^n$$

where SS consists of source terms as well as any other terms which can not be included in the other coefficients defined above.